

Farid Meziane
Elisabeth Métais (Eds.)

LNCS 3136

Natural Language Processing and Information Systems

9th International Conference on Applications
of Natural Language to Information Systems, NLDB 2004
Salford, UK, June 2004
Proceedings



Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

New York University, NY, USA

Doug Tygar

University of California, Berkeley, CA, USA

Moshe Y. Vardi

Rice University, Houston, TX, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Farid Meziane Elisabeth Métais (Eds.)

Natural Language Processing and Information Systems

9th International Conference on Applications
of Natural Language to Information Systems, NLDB 2004
Salford, UK, June 23-25, 2004
Proceedings



Springer

Volume Editors

Farid Meziane

University of Salford, School of Computing, Science and Engineering

Newton Building, Salford M5 4WT, UK

E-mail: f.meziane@salford.ac.uk

Elisabeth Métais

Laboratoire CEDRIC, CNAM

292 rue Saint Martin, 75141 Paris cedex 3, France

E-mail: elsa@cnam.fr

Library of Congress Control Number: 3540225641

CR Subject Classification (1998): H.2, H.3, I.2, F.3-4, C.2

ISSN 0302-9743

ISBN 3-540-22564-1 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH

Printed on acid-free paper SPIN: 11019275 06/3142 5 4 3 2 1 0

Preface

Welcome to NLDB04, the Ninth International Conference on the Application of Natural Language to Information Systems, held at the University of Salford, UK during June 23-25, 2004. NLDB04 follows on the success of previous conferences held since 1995. Early conferences then known as Application of Natural Language to Databases, hence the acronym NLDB, were used as a forum to discuss and disseminate research on the integration of natural language and databases and were mainly concerned with natural language based queries, database modelling and user interfaces that facilitate access to information. The conference has since moved to encompass all aspects of Information Systems and Software Engineering. Indeed, the use of natural language in systems modelling has greatly improved the development process and benefited both developers and users at all stages of the software development process.

The latest developments in the field of natural language and the emergence of new technologies has seen a shift towards storage of large semantic electronic dictionaries, their exploitation and the advent of what is now known as the semantic web. Information extraction and retrieval, document and content management, ontology development and management and natural language conversational systems are becoming regular tracks in the last NLDB conferences.

NLDB04 has seen a 50% increase in the number of submissions and has established itself as one of the leading conferences in the area of applying natural language to information systems in its broader sense. The quality of the submissions and their diversity have made the members of the program committee work more than usual. 65 papers were submitted from 22 different countries. 29 were accepted as regular papers, while 13 were accepted as short papers. The papers were classified as belonging to one of these themes:

- Natural Language Conversational Systems
- Intelligent Querying
- Linguistic Aspects of Modeling
- Information Retrieval
- Natural Language Text Understanding
- Knowledge Bases
- Natural Language Text Understanding
- Knowledge Management
- Content Management

This year we were honored by the presence of our invited speaker Fabio Ciravegna from the University of Sheffield, United Kingdom. His lecture on “Challenges in Harvesting Information for the Semantic Web” was highly appreciated and initiated vivid discussions.

We are very thankful for the opportunity to serve as Program Chair and Conference Chair for this conference. However, the organization of such event is a collective effort and a team work. First of all we would like to thank the members of the Program Committee for the time and effort they devoted to the reviewing of the submit-

ted articles and to the selection process. My thanks go also to the additional reviewers for their help and support. We would like to take this opportunity to thank the local organizing committee, especially its chairman Sunil Vadera, for their superb work. We would like to thank Nigel Linge the head of the School of Computing Science and Engineering, Tim Ritchings the head of the Computer Science, Multimedia and Tele-communication discipline and Mr. Gary Wright from the External Relations Division for their help and support.

Obviously we thank the authors for their high quality submissions and their participation to this event and their patience during the long reviewing process.

June 2004

Farid Meziane
Elisabeth Métais

Organization

Conference Chair

Elisabeth Métais, Conservatoire National des Arts et Métiers de Paris, France.

Program Committee Chair

Farid Meziane, University of Salford, UK.

Program Committee

1. Diego Mollá Aliod, Macquarie University, Australia
2. Kenji Araki, Hokkaido University, Japan
3. Mokrane Bouzeghoub, PRISM, Université de Versailles, France
4. Fabio Ciravegna, University of Sheffield, UK
5. Gary A Coen, Boeing, USA
6. Isabelle Comyn-Wattiau, CEDRIC/CNAM, France
7. Antje Düsterhöft, University of Wismar, Germany
8. Günther Fliedl, Universität Klagenfurt, Austria
9. Alexander Gelbukh, Instituto Politecnico Nacional, Mexico
10. Nicola Guarino, CNR, Italy
11. Rafael Muñoz Guillena, Universidad de Alicante, Spain
12. Jon Atle Gulla, Norwegian University of Science and Technology, Norway
13. Harmain Harmain, United Arab Emirates University, UAE
14. Helmut Horacek, Universität des Saarlandes, Germany
15. Paul Johannesson, Stockholm University, Sweden
16. Zoubida Kedad, PRISM, Université de Versailles, France
17. Leila Kosseim, Concordia University, Canada.
18. Nadira Lammari, CEDRIC/CNAM, France
19. Winfried Lenders, Universität Bonn, Germany
20. Jana Lewerenz, sd&m Düsseldorf, Germany
21. Robert Luk, Hong Kong Polytechnic University, Hong Kong
22. Heinrich C. Mayr, Universität Klagenfurt, Austria
23. Paul McFetridge, Simon Fraser University, Canada
24. Elisabeth Métais, CEDRIC/CNAM , France
25. Farid Meziane, Salford University, UK
26. Luisa Mich, University of Trento, Italy.
27. Ana Maria Moreno, Universidad Politecnica de Madrid, Spain
28. Jian-Yun Nie, Université de Montréal, Canada
29. Manual Palomar, Universidad de Alicante, Spain
30. Odile Piton, Université Paris I Panthéon-Sorbonne, France
31. Reind van de Riet, Vrije Universiteit Amsterdam, The Netherlands

VIII Organization

32. Hae-Chang Rim, Korea University, Korea
33. Tim Ritchings, University of Salford, UK
34. Hongchi Shi, University of Missouri-Columbia, USA
35. Niculae Stratică, Concordia University, Canada
36. Vijay Sugumaran, Oakland University Rochester, USA
37. Veda Storey, Georgia State University, USA
38. Lua Km Teng, National University of Singapore, Singapore
39. Bernhard Thalheim, Kiel University, Germany
40. Babis Theodoulidis, UMIST, UK
41. Sunil Vadera, University of Salford, UK
42. Ronald Wagner, University of Linz, Austria
43. Hans Weigand, Tilburg University, The Netherlands
44. Werner Winiwarter, University of Vienna, Austria
45. Stanislaw Wrycza, University of Gdansk, Poland

Additional Reviewers

1. Farida Aoughlis, University of Tizi-Ouzou, Algeria
2. Andrew Basden, University of Salford, UK.
3. Frances Bell, University of Salford, UK
4. Maria Bergholtz, Stockholm University, Sweden
5. Aleksander Binemann-Zdanowicz, Kiel University, Germany
6. Antonio Ferrández, Universidad de Alicante, Spain
7. Laura Compoy-Gómez, University of Salford, UK
8. Fiedler Gunar, Kiel University, Germany
9. Maria Kutar, University of Salford, UK.
10. Didier Nakache, CNAM, France.
11. Alessandro Oltramari, CNR, Italy
12. Peggy Schmidt, Kiel University, Germany

Organizing Committee

1. Sunil Vadera (Chair), University of Salford, UK.
2. Farid Meziane, University of Salford, UK.
3. Samia Nefti, University of Salford, UK.
4. Mohamad Saraee, University of Salford, UK.
5. Edwin Mit, University of Salford, UK.

Organization Website and Contact

The conference was organized by the Computer Science Research Centre and the School of Computing, Science and Engineering at the University of Salford, United Kingdom. The website of the NLDB Conference is www.nlbd.org. For all information, contact Farid Meziane at F.Meziane@salford.ac.uk

Table of Contents

Regular Papers

Natural Language Conversational Systems

- A Natural Language Model and a System for Managing TV-Anytime Information from Mobile Devices 1

Anastasia Karanastasi, Fotis G. Kazasis, Stavros Christodoulakis

- State- and Object Oriented Specification of Interactive VoiceXML Information Services 13

Thomas Schwanzara-Bennoit, Gunar Fiedler

- Interpreting Semi-formal Utterances in Dialogs about Mathematical Proofs 26

Helmut Horacek, Magdalena Wolska

Intelligent Querying

- Event Ordering Using TERSEO System 39

Estela Saquete, Rafael Muñoz, Patricio Martínez-Barco

- The Role of User Profiles in Context-Aware Query Processing for the Semantic Web 51

Veda C. Storey, Vijayan Sugumaran, Andrew Burton-Jones

- Deriving FrameNet Representations: Towards Meaning-Oriented Question Answering 64

Gerhard Fliedner

- Lightweight Natural Language Database Interfaces 76

In-Su Kang, Seung-Hoon Na, Jong-Hyeok Lee, Gijoo Yang

- Ontology-Driven Question Answering in AquaLog 89

Vanessa Lopez, Enrico Motta

- Schema-Based Natural Language Semantic Mapping 103

Niculae Stratica, Bipin C. Desai

- Avaya Interactive Dashboard (AID): An Interactive Tool for Mining the Avaya Problem Ticket Database 114

Ziyang Wang, Amit Bagga

Linguistic Aspects of Modeling

Information Modeling: The Process and the Required Competencies of Its Participants	123
<i>P.J.M. Frederiks, T.P. van der Weide</i>	
Experimenting with Linguistic Tools for Conceptual Modelling: Quality of the Models and Critical Features	135
<i>Nadzeya Kiyavitskaya, Nicola Zeni, Luisa Mich, John Mylopoulos</i>	
Language Resources and Tools for Supporting the System Engineering Process	147
<i>V.O. Onditi, P. Rayson, B.Ransom, D. Ramduny, Ian Sommerville, A. Dix</i>	
A Linguistics-Based Approach for Use Case Driven Analysis Using Goal and Scenario Authoring	159
<i>Jintae Kim, Sooyong Park, Vijayan Sugumaran</i>	

Information Retrieval

Effectiveness of Index Expressions	171
<i>F.A. Grootjen, T.P. van der Weide</i>	
Concept Similarity Measures the Understanding Between Two Agents ...	182
<i>Jesus M. Olivares-Ceja, Adolfo Guzman-Arenas</i>	
Concept Indexing for Automated Text Categorization	195
<i>José María Gómez, José Carlos Cortizo, Enrique Puertas, Miguel Ruiz</i>	

Natural Language Text Understanding

Acquiring Selectional Preferences from Untagged Text for Prepositional Phrase Attachment Disambiguation	207
<i>Hiram Calvo, Alexander Gelbukh</i>	
Semantic Enrichment for Ontology Mapping	217
<i>Xiaomeng Su, Jon Atle Gulla</i>	
Testing Word Similarity: Language Independent Approach with Examples from Romance	229
<i>Mikhail Alexandrov, Xavier Blanco, Pavel Makagonov</i>	

Knowledge Bases

Language Modeling for Effective Construction of Domain Specific Thesauri	242
<i>Libo Chen, Ulrich Thiel</i>	

Populating a Database from Parallel Texts Using Ontology-Based Information Extraction	254
<i>M.M. Wood, S.J. Lydon, V. Tablan, D. Maynard, H. Cunningham</i>	

A Generic Coordination Model for Pervasive Computing Based on Semantic Web Languages	265
<i>Amine Tafat, Michele Courant, Beat Hirsbrunner</i>	

Natural Language Text Understanding

Improving Web Searching Using Descriptive Graphs	276
<i>Alain Couchot</i>	

An Unsupervised WSD Algorithm for a NLP System	288
<i>Iulia Nica, Andrés Montoyo, Sonia Vázquez, Mª Antònia Martí</i>	

Enhanced Email Classification Based on Feature Space Enriching	299
<i>Yunming Ye, Fanyuan Ma, Hongqiang Rong, Joshua Huang</i>	

Synonymous Paraphrasing Using WordNet and Internet	312
<i>Igor A. Bolshakov, Alexander Gelbukh</i>	

Knowledge Management

Automatic Report Generation from Ontologies: The MIAKT Approach	324
<i>Kalina Bontcheva, Yorick Wilks</i>	

A Flexible Workbench for Document Analysis and Text Mining	336
<i>Jon Atle Gulla, Terje Brasethvik, Harald Kaada</i>	

Short Papers

Content Management

Towards Linguistic Foundations of Content Management	348
<i>Gunar Fiedler, Bernhard Thalheim</i>	

Constructing Natural Knowledge Ontologies to Implement Semantic Organizational Memory	354
<i>Laura Campoy-Gomez</i>	

Improving the Naming Process for Web Site Reverse Engineering	362
<i>Sélima Besbes Essanaa, Nadira Lammari</i>	

On Embedding Machine-Processable Semantics into Documents	368
<i>Krishnaprasad Thirunarayan</i>	

Information Retrieval

Using IR Techniques to Improve Automated Text Classification	374
<i>Teresa Gonçalves, Paulo Quaresma</i>	
Architecture of a Medical Information Extraction System	380
<i>Dalila Bekhouche, Yann Pollet, Bruno Grilheres, Xavier Denis</i>	
Improving Information Retrieval in MEDLINE by Modulating MeSH Term Weights	388
<i>Kwangcheol Shin, Sang-Yong Han</i>	

Identification of Composite Named Entities in a Spanish Textual Database	395
<i>Sofía N. Galicia-Haro, Alexander Gelbukh, Igor A. Bolshakov</i>	

Intelligent Querying

ORAKEL: A Natural Language Interface to an F-Logic Knowledge Base	401
<i>Philipp Cimiano</i>	

Accessing an Information System by Chatting	407
<i>Bayan Abu Shawar, Eric Atwell</i>	

Ontology-Based Question Answering in a Federation of University Sites: The MOSES Case Study	413
<i>P. Atzeni, R. Basili, D.H. Hansen, P. Missier, Patrizia Paggio, Maria Teresa Pazienza, Fabio Massimo Zanzotto</i>	

Semantic Tagging and Chunk-Parsing in Dynamic Modeling	421
<i>Günther Fliedl, Christian Kop, Heinrich C. Mayr, Christian Winkler, Georg Weber, Alexander Salbrechter</i>	

Semantic Filtering of Textual Requirements Descriptions	427
<i>Jorge J. García Flores</i>	

Author Index	435
-------------------------------	-----

A Natural Language Model and a System for Managing TV-Anytime Information from Mobile Devices

Anastasia Karanastasi, Fotis G. Kazasis, and Stavros Christodoulakis

Lab. of Distributed Multimedia Information Systems / Technical University
of Crete (MUSIC/TUC)

University Campus, Kounoupidiana, Chania, Greece
`{allegra,fotis,stavros}@ced.tuc.gr`

Abstract. The TV-Anytime standard describes structures of categories of digital TV program metadata, as well as User Profile metadata for TV programs. In this case study we describe a natural language model and a system for the users to interact with the metadata and preview TV programs stored in remote databases, from their mobile devices contrary to their limited configurations. By the use of the TV-Anytime metadata specifications the system limits greatly the possibility for ambiguities. The interaction model deals with ambiguities by using the TV-Anytime user profiles and metadata information concerning digital TV to rank the possible answers. The interaction between the user and the system is done by the use of a PDA and a mobile phone with metadata information stored on a database on a remote TV-Anytime compatible TV set.

1 Introduction

The number of digital TV channels has increased dramatically the last few years, and several industrial sectors and content producing sectors are active in defining the environment in which the TVs of the future will operate.

The TV-Anytime Forum is an association of organizations which seeks to develop specifications to enable audio-visual and other services based on mass-market high volume digital storage in consumer platforms - simply referred to as local storage [1]. These specifications target interoperable and integrated systems, from content creators/providers, through service providers, to the consumers and aim to enable applications to exploit the storage capabilities in consumer platforms. The basic architectural unit is an expanded TV set (known as a Personal Digital Recorder – PDR) capable of capturing digital satellite broadcasts according to user interests as they are described in his profile and storing them into large storage devices. The current TV-Anytime standard specifications define the structures for the metadata that can be used to describe TV programs and broadcasts, as well as for the metadata that can be used to describe the user profile. Expanded versions of the TV-Anytime architecture foresee also last mile TV-Anytime servers, Internet connection of the TV set and mobility aspects. Mobile devices (mobile phones, PDAs, etc.) in the TV-Anytime architecture can be used by a user to communicate with the home TV set not only for viewing TV programs, but also for managing the contents of the TV set (like previewing its con-

tents, searching for content, deleting content that has been recorded for him by the TV set, etc.) and for managing his profile preferences [2].

There is a strong need for new interface paradigms that allow the interaction of naïve users with the future TV sets in order to better satisfy their dynamic preferences and access information. The usual pc-based interfaces are not appropriate to interact with mobile devices (like mobile phones or PDAs) or with TV sets. Natural language interfaces (NLIs) are more appropriate interface styles for naïve users, and they can also support voice-based interactions for mobile devices.

The appeal of natural language interfaces to databases has been explored since the beginning of the '80s [6], [7]. Significant advances have been made in dialogue management [3], [4], [5], but the problem of reliable understanding a single sentence has not been solved. In comparison to the efforts made several years ago to enrich the databases with NLIs which faced the prohibitive cost of dialogues to fully clarify the query [3], our environment is more concrete than general purpose interfaces to database systems, since the structure imposed by the TV-Anytime specifications for the metadata greatly limit the possibilities for ambiguities.

The importance of natural language interfaces to databases has increased rapidly the last few years due to the introduction of new user devices (including mobile devices such as PDAs and mobile phones) for which traditional mouse based interfaces are unacceptable. Research has been published in the area of NLIs to interactive TV based information systems [8], [9]. A well-known problem with the NLIs is that user interactions may be ambiguous. Ambiguity in the NLIs is a serious problem and most systems proposed in the literature often lead to lengthy clarification dialogues with the user to resolve ambiguities [14]. These dialogues systems face the problem that the users often do not know the answers to questions asked by the system. Unlike the previous systems we do not resolve the remaining ambiguities with clarification. Instead we can take advantage of the TV-Anytime user profile specifications in order to rank the possible interpretations and present to the user at the top position the one with the highest ranking.

In this paper we present a model for natural language interactions with a TV set in an environment that follows the TV Anytime specifications, both for the TV program metadata as well as for the user profile metadata. The metadata are stored in databases with last mile connections. The natural language interactions are used to preview programs or summaries of programs as well as to completely manage the metadata and the programs that the TV set keeps for the user. In addition we describe an implementation of this TV-Anytime compatible natural language interaction model that works on a PDA and a mobile phone, which communicates with the TV-Anytime TV set for managing its programs and metadata and also allowing the previewing of TV programs from the mobile device.

The best-known dialogue systems that have been developed for digital TV and mobile environments are related to the MIINA project [11] and the Program Guide Information System of NOKIA [12]. In the context of MIINA project, a system has been developed for information retrieval from the set-top-box Mediaterminal of NOKIA. The user is allowed to insert queries for TV programs, channels, program categories and broadcast time, using a natural language. However, the natural language interaction in this model is rather simple since it is only related to the information provided by a traditional TV-Guide. The Program Guide Information System is an electronic call-in demo application offering information about television programs

over the phone by allowing the user to converse with the system in natural language sentences. This system is not based on TV-Anytime metadata structures for describing the programs or the user profiles. The scope of the interaction does not include any management of the stored content except retrieval or the user profiles. The main differences between those systems and the one described in this paper is that the present system uses the TV-Anytime content and consumer metadata specifications for a complete management of TV programs and user profiles, and that the system uses additional information that exists in the TV-Anytime User Profile in order to avoid length clarification dialogues and help the user to get the most relevant answers at the top of the result list.

In section 2 of this paper the natural language model for digital TV environment is presented, along with the functionality provided and the representation of the information that the system collects from the user's input. In section 3 we present the algorithm for resolving the ambiguities instead of using clarification dialogues. In section 4 there is the analysis of the system architecture and of the modules that constitute it. Section 5 presents the implementation environment of the system and of the applications from the client side. In section 6 we present an example of a user's utterance and the actions taken by the system in order to satisfy the user's request. Finally section 7 presents the results of the system's evaluation based on user experiments and section 8 concludes by summarizing the content of this paper.

2 The Natural Language Model for the Digital TV Environment

The proposed Natural Language Model allows a user to determine the rules of management of digital TV data (programs and metadata), retrieve TV program content based on any information of its metadata description, express his preferences for the types of TV programs that will be stored, manage his selection list (i.e. programs that have been selected by the PDR or the user himself as candidates for recording), by creating his profile and modify any of the above.

The user's utterance is constituted by a combination of sub-phrase. The categories of these sub-phrases are Introduction phrases, to define the functionality, Search phrases, to define the TV-Anytime information, Target phrases, to define where each of the functions is targeting, Temporal phrases , to define phrases about date and time and Summary phrases, to define summaries with audio/visual content.

The structure that represents the information gathered by the user's utterance is shown in figure 1. This structure consists of three parts namely Element, Element Type and Element Value. The first structure part (Element) is used to differentiate the TV-Anytime metadata information (modeled as *TVA-properties*) from the information that directs the system to the correct management of the user's input (modeled as *flags*). The TV-Anytime information about date and time is modeled as *temporal* Elements. The second structure part (Element Type) is used in order to further specialize the aforementioned information and to obtain its corresponding Element Value (the third structure part), from the user's utterance. When a user inserts an utterance into the system, it generates a feature structure [10] that follows the structure of the model.

Element	Element Type		Element Value
flags	action		retrieve insert delete profile
	target		box list profile
temporal	time		1 ... 24
	Day		Monday ... Sunday
	month		January ... December
	year		<YYYY>
	before	time	1 ... 24
		day	Monday ... Sunday
		month	January ... December
		year	<YYYY>
	after	time	1 ... 24
		day	Monday ... Sunday
		month	January ... December
		year	<YYYY>
	time indicator		pm or am
	day indicator		weekly
TVA-properties	genre		<list of genres from the TVA Specification>
	title		string of arbitrary length
	keyword		string of arbitrary length
	creator		string of arbitrary length
	name		string of arbitrary length
	country		string of arbitrary length
	date period		no value
	language		string of arbitrary length
	dissemination date		no value
	dissemination location		string of arbitrary length
	dissemination source		string of arbitrary length
	type		audio visual textual
	theme		string of arbitrary length
	format		characters frames minutes seconds
	length		string of arbitrary length
	minlength		string of arbitrary length
	maxlength		string of arbitrary length

Fig. 1. The structure of the natural language model

The ‘flags’ Element takes its value from the introduction phrases and the target phrases. The ‘TVA-properties’ Element takes its value from the search phrases and the summary phrases and the ‘temporal’ Element from the temporal phrases.

The feature structure can contain one, two or three Elements. These three types of feature structure are:

Type 1: Markers

- E.g. I want to see what is in my selection list

This utterance consists of an **introduction phrase** (I want to see what is) and a **target phrase** (in my list). The Element Type action of the Element markers takes the value ‘retrieval’ (information that comes from the introduction phrase) and the Element Type target takes the value ‘list’ (information that comes from the target phrase).

Type 2: Markers – TVA-properties

- E.g. I would like you to show me movies starring Mel Gibson

This utterance consists of an **introduction phrase** (I would like you to show me) and a **search phrase** (movies starring Mel Gibson). The Element Type action of the Element markers obtains the value ‘retrieval’ (information that comes from the introduction phrase), the Element Type genre of the Element TVA-properties obtains the value ‘movies’, the Element Type creator takes the value ‘actor’ and the Element Type name takes the value ‘Mel Gibson’ (information that comes from the search phrase).

Type 3: Markers – TVA-properties – Temporal

- E.g. Insert English spoken mystery movies broadcasted at midnight into my selection list.

This utterance consists of an **introduction phrase** (Insert), a search phrase (English spoken mystery movies broadcasted), a **temporal phrase** (at midnight) and a **target phrase** (into my selection list). In this case, the Element Type action of the Element markers takes the value ‘insert’ (information that comes from the introduction phrase), the Element Type target takes the value ‘list’, from the target phrase, the Element Type genre of the Element TVA-properties takes the values ‘mystery’ and ‘movies’, the Element Type language takes the value ‘English’ and in the feature structure there is also the Element Type dissemination value, but without value. This information comes from the search phrase. Also, in the Element temporal, the Element Type time takes the value ‘24’ and the Element Type time indicator takes the value ‘am’. This information also comes from the search phrase.

The TV-Anytime metadata model integrates specifications for content metadata used to describe digital TV Programs in terms of various features and specifications for user preferences used to filter program metadata. These user preferences are modelled by the FilteringAndSearchPreferences Descriptor Scheme (DS) and the BrowsingPreferences DS. The FilteringAndSearchPreferences Descriptor Scheme (DS) specifies a user’s filtering and/or searching preferences for audio-visual content. These preferences can be specified in terms of creation-, classification- and source-related properties of the content. The FilteringAndSearchPreferences DS is a container of CreationPreferences (i.e. Title, Creator), ClassificationPreferences (i.e. Country, Language) and SourcePreferences (i.e. DisseminationSource, DisseminationLocation). The BrowsingPreferences DS is used to specify a user’s preferences for navigating and accessing multimedia content and is a container of SummaryPreferences (i.e. SummaryType, SummaryTheme, SummaryDuration) and PreferenceCondition (i.e. Time, Place).

For the retrieval of the personalized content metadata, the management of the personalized content and the creation of the user’s profile, the utterance contains in its body one or more search phrases. The system will create a TV-Anytime XML document, compatible with the UserIdentifier and the FilteringAndSearchPreferences

Descriptor Schemes of the TV-Anytime metadata specification. For the forth system function, the definition of the user's preferences for the characteristics of an audio-visual content summary, the system constructs a TV-Anytime XML document, compatible with the UserIdentifier and the BrowsingPreferences Descriptor Schemes, with values in the fields of the SummaryPreferences and the PreferenceCondition (for handling the time of the summary delivery).

A user's selection list is a list of information about program's metadata that the system recommends to the user based on his preferences expressed either at his TV-Anytime profile or directly by him. Every program in this list has a status. The four possible values of this status are: undefined, toBeRecorded, recorded, toBeDeleted. So, if the user wants to manage the contents of his selection list or the list of his stored contents the actions that take place are represented in figure 2:

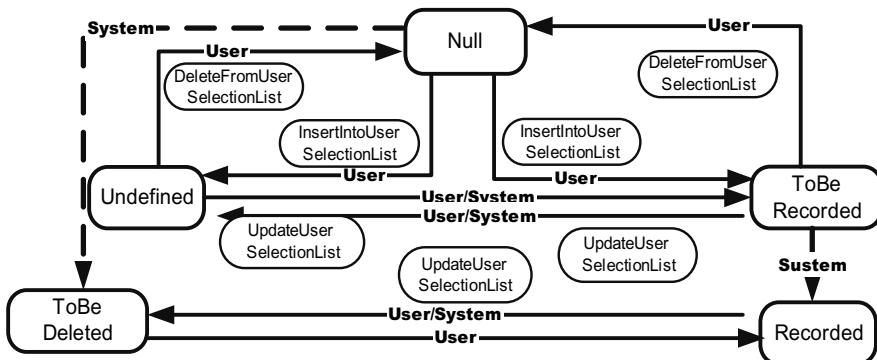


Fig. 2. The state machine for managing the status of a program in the user's selection list

3 Resolving Ambiguities – The Algorithm

From the user's utterance the system repossess the TV-Anytime category he is referring to. If there is no such feedback from the utterance the system, by following a specific number of steps, tries to resolve the ambiguity. First, it collects word by word the sub-phrase with the ambiguities and creates a table of these words. Then -by using a stop list containing words with no semantic values, such as prepositions, pronouns, conjunctions, particles, articles, determiners and so on- it eliminates the words that match any word in this list. However, the system retains the existence of an 'and' or an 'or' for the optimum cross-correlation of the results. Then the system gathers the remaining words and by filtering them through its database or other existing ontologies it returns a TV-Anytime XML document that is compatible with the FilteringAndSearchPreferences DS. This descriptor scheme is the one that is used for the semantic resolving of the words. Finally, the system checks for matches in any existing TV-Anytime user's profile. The classification of the results is important in order to prioritize the user's preferences.

Algorithm for Ambiguities Resolution

```

Subroutine: semantic resolver
  for every word with ambiguity check the stop list
  if there is no match.
    check for TVA semantics
    if there are TVA semantics add a specific weight value
    if there is a user profile
      check from semantics from profile
      if there is a match
        add a weight value based on the preference value from profile to the TVA
        semantics
    if there is a match
      cut the word
    return

  check the words with ambiguities for an 'and' or an 'or'
  if there is no match
    call semantic resolver
    check for same strings that contain words with the same semantic
    cut the rest
    search for results
    rank the results based on the same TVA semantic
  else
    call semantic resolver
    search for results
    group the results based on the same TVA semantic
    rank the results based on the weight value

```

4 System Architecture

In figure 3 we present the overall architecture of the natural language system.

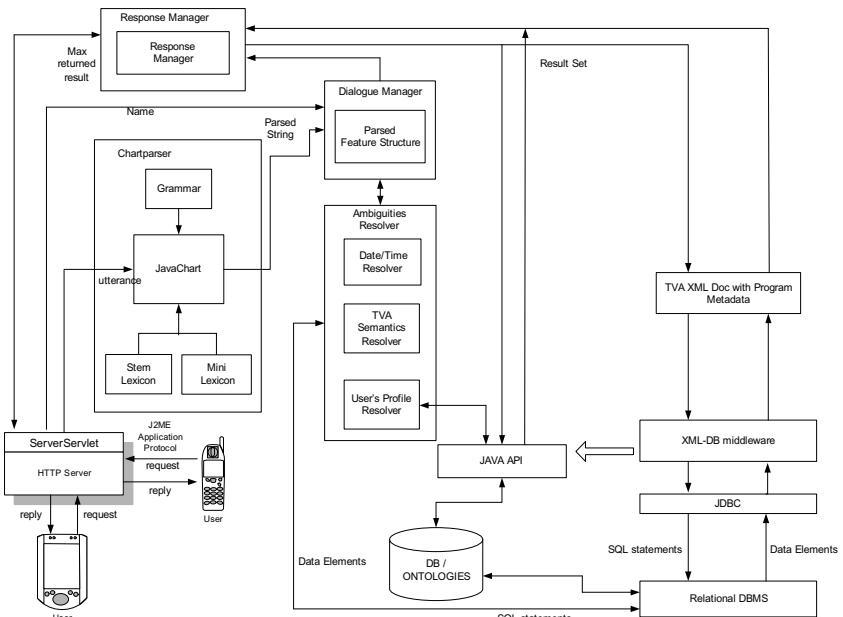


Fig. 3. The Natural Language System Architecture

The insertion of the utterance is made by the use of a wireless device (mobile phone, PDA). Then it is forwarded to the **ChartParser module**. The ChartParser module consists of the JavaChart parser [13] that creates a feature structure with the information from the user's input. There exist two lexicons, the stem lexicon, which contains the stems of the words used in this language, and the words that help to give the right values to the TV-Anytime categories, and the mini lexicon, which contains the endings of the words. Finally, there is the grammar that follows a unified-based formalism.

The **Dialogue Manager** acts as the core module of the system. It is responsible for communicating with all the other system modules, to create a proper structure of the feature structure so the other modules to extract the information, to deal with ambiguities by communicating with the **Ambiguities Resolver** module, to interact with the server to retrieve the results and to pass the information to the **Response Manager** module for the conduction of a proper message for the user. It takes as an input a list of feature structures from the chart parser and the user's data from the application that he/she is using. It checks for specific properties in order to eliminate the list of the feature structures and, in the case there are any ambiguities, it passes to the Ambiguities Resolver module the list of the words with the ambiguities. Finally, it creates a structure with information about the action, the target and the TV-Anytime XML document from the user's input.

The content management system architecture follows a multi-tier approach and consists of three tiers. The lowest tier handles the metadata management. The middleware tier includes all the logic for interfacing the system with the outside world. The application tier enables the exchange of information between the server and heterogeneous clients through different communication links. The **Relational Database** of the system contains the TV-Anytime Metadata information, as well as a number of **ontologies**, with information concerning digital TV applications. The **Relational DBMS** manages the transactions, utilizes a **Java API** (implemented for the extraction of the functionality for filtering, retrieval and summarization) and cooperates with the **XML-DB middleware**. The **XML-DB middleware** is a set of software components responsible for the management of the TV-Anytime XML documents and the correspondence of the TV-Anytime Metadata XML schema with the underlying relational schema.

The **Ambiguities Resolver** module consists of three modules that are responsible for the resolution of different kinds of ambiguities. The *Date/Time Resolver* is the component that converts the temporal phrases in a TV-Anytime compliant form. The *TVA Semantics Resolver* communicates with the relational DBMS and is responsible to attach TV-Anytime semantics to the words with the ambiguities. We use the *User's Profile Resolver* to help the ranking of the final results. Every user can have one or more User Profiles, which represents his interests. The module filters the list of the words from any existing user's profile and returns a FilteringAndSearchPreferences XML document with values from the corresponding TV-Anytime categories. Finally, it passes this document to the Response Manager module.

The **Response Manager** module interacts with the system's database, by providing it the structured information, executes the appropriate functions, retrieves the results and classifies them accordingly. Then, it creates a message and adds it to any existing result list. The user must understand from the message what exactly the system done to satisfy his request and get an error message if something went wrong.

5 Implementation Environment

The implementation platform consists of the MySQL Platform [15]. The implementation of the server was based on Java 2 and the class files were compiled using JDK1.4.1. [16]. The parsing of the user's utterance was made by the use of the JavaChart parser, a chart parser written in Java. For the implementation of the communication between the server and the client, we have exploited the JAVA Serlvet technology in the server side by developing a servlet that acts as the interface between the user client and the database server or the PDR interface. This servlet was locally deployed for testing on an Apache Tomcat v4.0.1 server and the class files were compiled using JDK1.4.1.

Two cases are considered related to the wireless device used on the system's client side. The first one is the application that runs on any Java-enabled mobile phone device and the second is the application that runs on a PDA device. For both cases of the remote access application, the client establishes an http connection with the server.

For the client (mobile device) the implementation platform consists of the Java 2 Platform, Micro Edition (J2ME) [16]. The Connected Limited Device Configuration (CLDC) has been used for the limited capabilities of the mobile phone. The Mobile Information Device profile (MIDP) is an architecture and a set of Java libraries that create an open, third party application development environment for small, resource-constrained, devices.

For the second implementation of the client side we used JEODE Java Runtime Environment [16] for the PDA device client. The JEODE runtime environment supports the CDC/Foundation Profile and the Personal Profile J2ME specifications that support implementations in PersonalJava and EmbeddedJava.

6 Evaluation

This section presents some preliminary results of the system's evaluation. The evaluation has been based on a user experiment that was performed by ten laboratory users. All the users had previous experience using computers and graphical interfaces. Nevertheless none of them had ever used a natural language system before. There were three main tasks used in the evaluation. The first one was for the users to use the system to define their user profile, the second one to interact with the system by using a set of utterances with no ambiguities and the third one to interact with the system by using a set of 10 utterances with ambiguities.

The functionality provided by the natural language system was also provided by the use of alternative menu-driven user interfaces for wireless devices (mobile phone, PDA). Based on the preliminary results it becomes clear that the end users found the system easier to use with the NLI than with the traditional user interfaces. The NLI was shown to provide an easy way to specify TV-Anytime structures without complex navigation between screens. For utterances that showed no ambiguities the system proved to fully exploit the structured TV-Anytime model capabilities for filtering in order to retrieve the qualified ranked results.

In the case of the utterances with ambiguities, we have considered 100 interactions in total. All the utterances contained a sub-phrase with no declaration of any TV-Anytime categories. For example, two of the utterances considered were:

- I want comedies with Tom English
- Record movie from Russia with love

In order to evaluate our approach for the use of the existing user profiles in order to rank the possible answers in case of ambiguities we have considered different types of user profiles so that the similarity between the user's preferences in these profiles and the specific input utterances to be near 0,5.

Diagram 1 shows the number of interactions per rank position for the cases that the system has either used the user profile in order to better rank the results or not. When the system uses the user profile to rank the results, we get about 90% of the exact results in the first 20 positions of the ranked resulting lists. This percentage varies according to the result list. In the first 5 positions we get the 65% of the exact results. When the system does not use the user profile to rank the results, we get about 80% of the exact results in the first 20 positions of the ranked resulting lists. In the first 5 positions we get the 40% of the exact results.

Diagram 2 shows the number of interactions in the top percentage of the total number of results. When the system uses the user profile to rank the results we get about 90% of the exact results in the 40% of the total number of results. In the case that the system does not use the user profile we get about 90% of the exact results in the 70% of the total number of results.

More systematic and larger scale evaluation work is still under way.

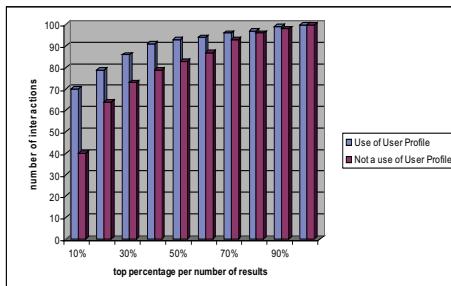


Diagram 1. Number of interactions per rank position

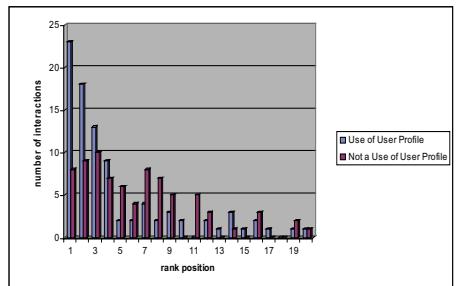


Diagram 2. Number of interactions per top percentage of the total number of results

7 Summary – Conclusions

In this paper we described the design and the implementation of a natural language model for managing TV-Anytime information (program metadata, user profile metadata) which are stored in databases in home TV-Anytime boxes or in last mile TV-Anytime servers. The NLIs allow the users to manage TV-Anytime metadata from PDAs and mobile phones in order to determine the rules of management of digital TV data (programs and metadata), retrieve TV program content based on any information of its metadata description, express his preferences for the types of TV programs that

will be stored, manage his selection list (i.e. programs that have been selected by the PDR or the user himself as candidates for recording) or modify any of the above.

The natural language model was developed to be compatible with the TV-Anytime specifications and manages the information of the content metadata categories for TV programs and the user's profile metadata. It is expandable to future additions in the TV-Anytime metadata specifications.

In order to satisfy the proposed functionality a dialogue model was developed that contains: Introduction phrases, to define the functionality, Search phrases, to define the TV-Anytime information, Target phrases, to define where each of the functions is targeting, Temporal phrases, to define phrases about date and time and Summary phrases, to define summaries with audio/visual content.

The structures defined for the metadata by the TV-Anytime standard limit significantly the possibility for ambiguities in the language. Thus most queries are answered precisely from the underlined database system. Whenever ambiguities occur, the system firstly checks for specific words in the utterance, then searches existing ontologies and attaches semantics to every word that appears with ambiguity and finally checks the TV-Anytime user's profile to attach a weight value to the search results. The algorithm checks for the best cross-correlations and unifies the results by assigning the regular weight values to the results.

The implementation of the natural language model runs on a mobile phone and a PDA. Preliminary evaluation studies have shown it to be a good tool for these environments, better than traditional PC interfaces. Larger scale experimentation is currently underway. In addition the integration and interaction with domain specific ontologies related to TV programs [17], [18]. Our current research aims to show that natural language (and speech) interfaces are appropriate interface styles for accessing audiovisual content, stored in home information servers, from mobile devices.

Acknowledgments. The work presented in this paper was partially funded in the scope of the DELOS II Network of Excellence in Digital Libraries [19].

Reference

1. The site of the TV-Anytime Forum, <http://www.tv-anytime.org>
2. Kazasis, F.G., Mousoutzis, N., Pappas, N., Karanastasi, A., Christodoulakis, S. (2003). *Designing Ubiquitous Personalized TV-Anytime Services*. In the International Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS), Klagenfurt/Velden, Austria.
3. Popescu, A., Etzioni, O., Kautz, H. (2003). *Towards a Theory of Natural Language Interfaces to Databases*. In Proceedings of the 8th International Conference on Intelligent user interfaces.
4. Core, M.G., Moore, J.D., Zinn, C. *Initiative in Tutorial Dialogue*. In Proceedings of ITS 2002 Workshop on Empirical Methods for Tutorial Dialogue Systems (ITS-02), 2002.
5. Sattingh, S., Litman, D., Kearns, M., Walker, M., *Optimizing Dialogue Management With Reinforcement Learning: Experiments with the NJFun System*. In Journal of Artificial Intelligence research (JAIR), 2002.
6. Hendrix, G., Sacerdoti, E., Sagalowicz, D., Slocum, J., *Developing a natural language interface to complex data*. In ACM transactions on Database Systems 3(2), pages 105-147, 1978.

7. Androutsopoulos, I., Ritchie, G.D., Thanisch, P. *Natural Language Interfaces to Databases – An Introduction*. In Natural Language Engineering, vol. 1, part 1, pages 29-81, 1995.
8. Johansson, P., Degerstedt, L., Jönsson, A. (2002). *Iterative Development of an Information-Providing Dialogue System*. In Proceedings of the 7th Workshop on User Interfaces for All. Chantilly, France.
9. Ibrahim, A., Johansson, P. (2002). *Multimodal Dialogue Systems for Interactive TV Applications*. In Proceedings of 4th IEEE International Conference on Multimodal Interfaces, Pittsburgh, USA. pp. 117-222.
10. Jurafsky, D., Martin, J.H. (2000). *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. New Jersey: Prentice Hall.
11. Multimodal Interaction for Information Appliances (MIINA Project),
<http://www.ida.liu.se/~nlp/MIINA/>
12. NOKIA – Talkative TV-Guide, Program Guide Information System,
<http://www.nokia.com/nokia/0,,27727.00.html>
13. Degerstedt, L. (2002). *JavaChart User Manual*, <http://nlpfarm.sourceforge.net/javachart/>
14. Maier E. (1997). *Clarification dialogues in VERBMOBIL*. In Proceedings of EuroSpeech-97, pp. 1891-1894, Rhodes, 1997
15. MySQL: The World's Most Popular Open Source Database, <http://www.mysql.com/>
16. Java Technology & J2ME, <http://java.sun.com/> , <http://java.sun.com/j2me/>
17. Tsinaraki, C., Fatourou, E., Christodoulakis, S. *An Ontology-Driven Framework for the Management of Semantic Metadata describing Audiovisual Information*. In Proceedings of CAiSE, Velden, Austria, 2003, pp 340-356
18. Tsinaraki, C., Polydoros, P., Christodoulakis, S. *Integration of OWL ontologies in MPEG-7 and TV-Anytime compliant Semantic Indexing*, CAiSE 2004 (accepted for publication)
19. DELOS Network of Excellence in Digital Libraries, <http://delos-now.iei.pi.cnr.it/>

State- and Object Oriented Specification of Interactive VoiceXML Information Services

Thomas Schwanzara-Benoit¹ and Gunar Fiedler²

¹ Computer Science Department, Databases and Information Systems Group
Brandenburg University of Technology at Cottbus,
PO Box 101344, 03013 Cottbus, Germany
tsb77@web.de

²Institute for Computer Science and Applied Mathematics
University Kiel, Olshausenstr. 40, 24098 Kiel, Germany
fiedler@is.informatik.uni-kiel.de

Abstract. Usually internet information services are based on HTML, but now more and more phone speech information services appear. In February 2003 VoiceXML 2.0 was published by the VoiceXML Forum to bring the advantages of web-based development and content delivery to interactive voice response information services. Such services include content generation, navigation and functionality support which has to be modeled in a consistent way. This document describes a SiteLang oriented specification approach based on media objects as well as movie concepts for story and interaction spaces. This results in a systematic simplification of the speech application development process. We have created a VoiceXML based information system prototype for an E-Government system which will be used as an example within this paper.

1 Introduction

Voice response information services differ from GUI based web services because they use natural speech technology and DTMF input instead of a computer with a web browser. In a normal man-machine conversation, the partners change their roles between speaking and listening to perform a dialog.

1.1 Voice XML Based Information Services

VoiceXML (VXML) is designed for creating audio dialogs that feature synthesized speech, digitized audio, recognition of spoken words and DTMF key input, recording of spoken input, telephony and mixed initiative conversations. VXML is a standard dialog design language that developers could use to build voice applications. As the dialog manager component it defines dialog constructs like form, menu and link, and the Form Interpretation Algorithm mechanism by which they are interpreted. A caller uses DTMF or speech as system input and gets synthetic speech or pre-recorded audio

as system output. Dialogs are added through the design of new VoiceXML documents which can be extended by a web application server with database connection. The architectural model assumed by this document has the following components:

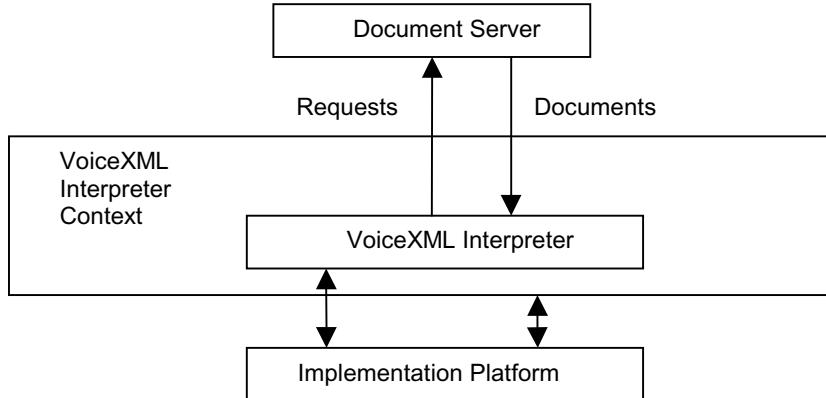


Fig. 1. VXML Architectural Model

A document server processes requests from a client application, the VXML interpreter, through the VXML interpreter context. The server produces VXML documents in reply, which are processed by the VXML interpreter. The VXML interpreter context may monitor user inputs in parallel with the VXML interpreter. For example, one VoiceXML interpreter context may always listen for a special escape phrase that takes the user to a high-level personal assistant, and another may listen for escape phrases that alter user preferences like volume or text-to-speech characteristics. The implementation platform is controlled by the VXML interpreter context and by the VXML interpreter. For instance, in an interactive voice response application, the VXML interpreter context may be responsible for detecting an incoming call, acquiring the initial VXML document, and answering the call, while the VXML interpreter conducts the dialog after answer. The implementation platform generates events in response to user actions and system events. Some of these events are acted upon by the VXML interpreter itself, as specified by the VXML document, while others are acted upon by the VXML interpreter context.

1.2 Problems with Voice Based Man-Machine Communication

Although interesting speech technology exists and already runs they are also drawbacks and problems.

Finances: telecommunication provider, telephony systems, speech recognition components, VXML browser and TTS-voices cost money

Naturalness: some users do not want to talk to a computer

Reliability: speech recognition components are not predictable and as good as a human in a call center

Development Process: systematic approaches for voice application specification and documentation are just in the beginning

Short dialogs and patience: questions and functionality should be as short and precise as possible because it takes times to listen to the system

Navigation: Caller has to navigate to desired functionality and data

Long dialogs: some users want to get a full information packet read by the system instead to navigate

VXML dialogs contain only data and sentences for what they are programmed for so the conversation is limited

We use of standard methods (object oriented, entity relationship), standard software (Java, MySQL, Apache, Tomcat) and systems (VoiceXML 2.0, OptimTalk) and we have developed a voice application system which is free and easy to use. Through the use of DTMF as primary input and speech recognition as secondary input any voice application is usable in many environments.

1.3 State- and Object Oriented Approach

A lot of approaches for the conceptual modeling of internet sites have been summarized and generalized in [ScT00]. This and other approaches can be adapted to the development of voice applications. Workflow approaches try currently to cover the entire behavior of systems. Wegner's interaction machines [GST00] can be used for formal treatment of information services. Semantics of information services can be based on abstract state machines which enable in reasoning on reliability, consistency and live ness [Tha00].

UML (Unified Modelling Language) is [For00] a standard notation for the modeling of real-world objects as a first step in developing an object-oriented design methodology. Its notation is derived from object oriented specifications and unifies the notations of object-oriented design and analysis methodologies. After the three gurus (Grady Booch, Ivar Jacobson and James Rumbaugh) finished creating UML as a single complete notation for describing object models, they turned their efforts to the development process. They came up with the Rational Unified Process (RUP), which is a general framework that can be used to describe specific development processes. We use the following UML concepts according to the RUP to specify our software application: use case, state, object, class and package.

HTML is [W3HTML4] the universally understood web language, similar to VXML and gives authors means to:

Publish online documents with text

Retrieve online information via hyperlinks, at the click of a button

Design forms for conducting transactions with remote services

We use HTML for the textual presentation and definition of the dialogs. As a result the development of voice applications can be reduced to graphical web development under consideration of voice aspects.

The software development starts with the specification of the story, processes and scenarios through requirement specification, process description and use cases. The software developer is free to use other modeling methods. Our use cases are based on story description and process description. With the help of this use cases we define state machines which represent the dialog and information flow. These FSMs are basis for the real spoken dialog, the navigation and the behavior of the application. Based on the dialog flow of the FSMs HTML pages are specified. As the last step of specification media objects will be defined and integrated in HTML. In the last step the HTML pages are translated into VXML pages.

Beside we explain the specification by introducing our speech application prototype SeSAM which was designed to support communal E-Government purposes. SeSAM supports the management of users, groups, messages and appointments.

2 Voice Application Specification

The implementation of voice services might be rather complex, so this specification approach aims to simplify the software development process of a voice application project. The voice software development process starts with the requirement analysis to get the real customer needs. On the base of the intuitive main objectives and under consideration of the user we develop structured use cases, finite state machines, a dialog specification and media objects.

2.1 Story Specification

A story describes the set of all possible intended interactions of users with the system. It is often defined generally in a product specification document. Additionally the story specification could be supported by a business process specification which describes the processes which should be supported by the software system.

A brief description of our E-Government SeSAM application as a story space could be the following:

```
Story {
    Visitor {Group Information, Public Messages, Public Political Appointments}
    RegisteredUser {Group Information, Messages, Appointments}
}
```

The system is used by registered users like party members or politicians and visitors who represent the inhabitants of a town. The users, or callers, will get information about members and groups, messages and appointments by phone. The proposed story specification method is not rigid, so other form could be used.

2.2 Process Specification

A process could be anything that operates for a period of time, normally consuming resources during that time and using them to create a useful result.

A process model is a general approach for organizing a work into activities, an aid to thinking and not a rigid prescription of the way to do things. It helps the software project manager to decide what work should be done in the target environment and in what sequences the work is performed. A process specification is useful for systems where voice interfaces are intended to support the processes or goals of the organization. An employee who answers customer questions by phone could be replaced by an automatic voice system which offers the desired information by structured FAQ. Process specification costs time for analysis and design, but it is a good documentation and a useful basis for further requirement specification.

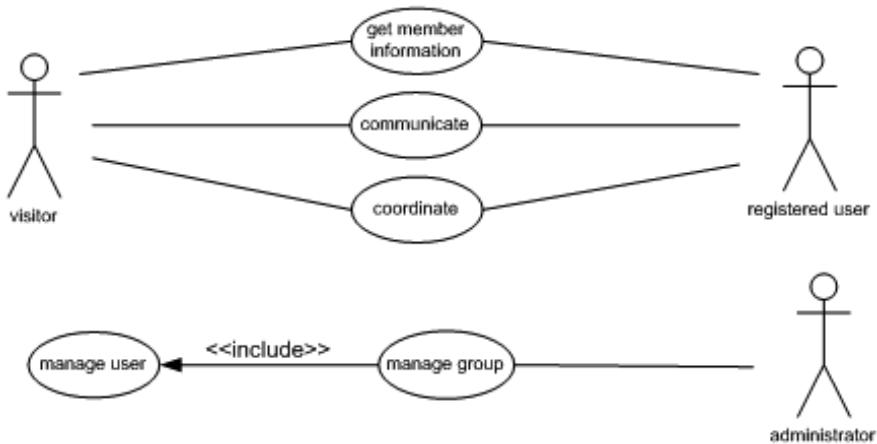


Fig. 2. Use Case Modeling

2.3 Use Case Specification

Based on intuitive goals, a story and a business process description we model the requirements of the application with use cases known from the Unified Modelling Language UML.

A use case describes a set of actions of a system watched by the user which causes some results. In our case it describes the set of scenarios and is started by an actor. A scenario is a specific set of actions that clarify the behavior of the voice software system and represent the basis for abstraction and specification. An actor represents one role outside of a system. Graphically a use case is presented though the form of an ellipse and is connected with an actor. It can be used by other use cases or uses other ones.

We use different levels of abstraction for use case modeling. There are other use case diagrams for the actions coordinate, communicate and get member information which should clarify and describe generally the interaction between the user and the system.

2.4 Dialog Structure Specification

The result of the story definition and requirement analysis is a specification of the typical interaction of a user with the computer voice system. But this documentation contains no information about navigation and the dialog structure of the VXML pages which contains the functionality and data. Based on the nature of a discussion each VXML page contains the data spoken by the TTS system and data about the reaction if the caller says something. DTMF for the selection of functionality or data is limited to the number set 0...9 and the special characters * and #.

We model navigation and the structure of the whole story and for each individual use cases with finite state machines $M = (F, A, Q)$ where:

1. Q is a finite set of conversation states
2. A is a finite set of user input symbols (user inputs like DTMF or speech or clicks)
3. F is a function of the form $F : Q \times A \rightarrow Q$

Every state of the FSM is regarded as a conversational state. In a conversational state the systems says something, asks the user something for an answer and goes to another conversational state, depending of user input. A transition from one state to another state defines the user input (DTMF, recognized speech command) in a certain state and defines the change into another conversational state.

Figure 3 shows the dialog structure specification for the use case communication of our SeSAM application. The system begins to read the start dialog at the start page and ask the caller what to do. The caller could decide by speaking or using DTMF to go to services where all information services will be read. From the service overview the caller could go to message services, selects there my messages, gets a message list and select then his desired message which will be read.

At the end of this phase you should have a generic FSM for the whole story and special FSM for every use case. All FSMs should be integrated into one FSM. The dialog structure specification as a FSM defines the navigation structure of the voice applica-

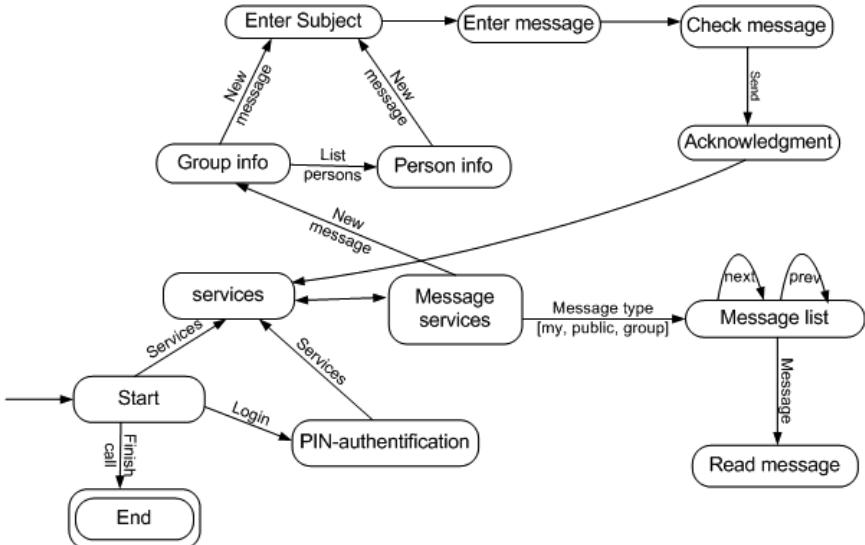


Fig. 3. Dialog Structure Specification through FSMs

tion but it contains no data about spoken sentences. This is done by a dialog step specification in HTML

2.5 Dialog Step Specification in HTML

The FSM specification is now added with data of spoken sentences and prompts through the HTML dialog step specification. An HTML page contains conversational data, prompts and answers for one conversational state and links to other conversational states. We map every state of a FSM to one HTML page. The next listing shows a dialog step definition of the state services from figure 3.

```

<html><head><title> Information Services </title></head>
<body>
    <p> You have selected information services.</p>
    <p> Please select now your desired service:</p>
    <ol>
        <li><a href="member.html">
            Member information </a></li>
        <li><a href="messages.html"> Messages </a></li>
        <li><a href="appointments.html">
            Appointments </a></li>
    </ol>
</body></html>

```

Every transition of the FSM (a user input which results in a new state) is mapped into a hyperlink to another page. It is free to the developer to use graphical mouse hyperlinks or JavaScript user input functions. On each page the machine says something and expects input from the user to go to another page. It is important to keep in mind that the specification is the basis for the voice application and text and menus should be as short and objective as possible. If there is any dynamically information, e.g. from a database, then it will be modeled with static example data.

At the end of the dialog step definition we have a HTML prototype which behaves similar to the VXML application. One advantage of this specification is the graphical definition of the dialogs, software developers are used to this. On the other hand you get a prototype as a set of HTML files which could be simulated to analyze the dialog and to improve the conversation.

2.6 Media Object Specification

The static graphical HTML-prototype of the voice application contains navigation structure and the 'adornment' of the scenes. It is filled with sample data and contains no dynamic information, e.g. from databases.

Now we design the media objects of our application to add dynamic functionality to our prototype. Generally, a media object is an interesting object for the application, e.g. messages, appointments, groups or users. From the object oriented point of view a media object is the member of a class with certain attributes and methods. From the database view a media object could be a view on a database.

We used the following approach to design our media objects:

1. Find nouns which are used in story, use cases, FSM and in the HTML pages
2. Identify methods which are used in the HTML pages and create object names
3. Categorization of the nouns, identify generalizations
4. Remove nouns and concepts which does not name individual concepts
5. Choose short class names for media objects and document each one shortly
6. Identify attributes, associations, relations ...

In our application the use case communication uses the media object message. This media object message will be used in almost all states (and therefore all HTML pages) for communication. In the state messageList the media object message provides a function like getMessageList(String messageType) where a certain count of messages will be displayed in HTML or read in VXML.

2.7 Integration of HTML and Media Objects

The speech application development process is hard to overview; therefore we develop at first a dynamic HTML voice prototype from our static HTML prototype which has the 'same' behavior and functionality as the VXML voice application.

The integration of HTML and media objects is done by inserting the media objects with their functionality into the HTML pages via special tags. At this point you have to choose a programming language and you have to implement the functions of your class. The following listing shows HTML code within a Java Server Page which is enriched by the media object message.

```
<jsp:useBean id="message" class="SeSAM.HTML.Message">
<% message.setSession(session); %>
<html>
<head><title> Message list </title></head>
<body>
<p> You are in the <%= message.type%> overview </p>
<p> Please select now your desired message </p>
<ol>
    <%if (message.type.equals("public") ||
        (message.type.equals("my"))
        out.println(message.getMessageList()); %>
    <li><a href="messages.html"> back to messages
    </a></li>
    <li><a href="services.html"> back to services
    </a></li>
</ol>
</body></html>
```

The listing shows HTML code with JSP scripting elements. Other object oriented programming languages like PHP or C++ could also be used for implementation.

At the end of this phase we have a dynamic graphical HTML prototype which has the same dialog flow and functionality as the VXML application and which uses the same data from the database. At the end of this phase the HTML web application has to be simulated and reviewed under consideration of conversational and speech aspects.

2.8 HTML to VXML Translation

In this step each HTML page is translated to a VXML page. The VXML page could use the same media objects, but some output functions have to be modified or added to get VXML output instead of HTML output. This process can be automated by some software tools.

The result of this translation is a ready VXML application which could be easily run on a normal PC or a VXML based telecommunication system. Future work tends to automate this translation process; a specification in XML for the automatic generation of HTML and VXML could decrease the development time.

The following listing shows a typical SeSAM E-Government VXML document which is similar to a typical HTML page and which uses the dynamic object member. It has to be mentioned that the whole application could be tested without media object tags, too.

```

<jsp:useBean id="member" class="SeSAM.VXML.Member">
<% member.setSession(session);%>
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <block>You chose member information services</block>
    <field name="choiceMember">
      <prompt> Please choose the desired group or other
              services </enumerate></prompt>
      <option dtmf=<%= member.getDTMFcounter()%>
              value="services"> back to services </option>
    <%= member.getTopGroupsOptions()%>
    <filled>
      <%= member.getTopGroupsFilledIfs()%>
      <if cond="choiceMember=='services'">
        <goto next="../servlet/services"/>
      </if>
    </filled>
  </field>
</form>
</vxml>

```

2.9 Application Simulation

The simulation and run of the application is important to validate and check the behavior, the data and the functionality of the system.

We suggest the following points to perform rapid development and validations:

1. Specify and test FSMs for navigation and input validation
2. Specify and test a static HTML dialog for a use case
3. Specify and test a static VXML dialog for the use case
4. Add dynamic media objects to HTML files and test them
5. Add dynamic media objects to VXML files and test them in a web browser with XML support and XML validation (Mozilla, IE)
6. Test each VXML files in your voice browser (OptimTalk, Elvira)
7. Test the whole use case without graphical user interface (from login to the end of the call)
8. Call the voice application on a voice platform by phone

Application simulation starts in the design phase. Any FSM, static HTML pages and static VXML pages should be simulated with suitable tools to validate the design and the specification of the application.

At any time it is important to interact with the VXML application and to simulate it - without a graphical interface. The tester has to listen to it because the real application environments are (mobile) phones or IP-telephony. Through a VXML-simulation you can also validate the FSM, HTML and media object specification because the VXML programmed code is based on these specifications.

3 Summary

In this work we present a specification for the rapid development of voice application services. The specification is part of the documentation and basis for implementation. Due to the co-design of a HTML prototype and a VXML application the specification is easily understandable and is based on a robust finite state machine navigation structure. Media objects with the according database tables can be added in any object oriented programming language. Besides the specification approach speeds up implementation.

3.1 Voice Application Development Process

The application specification is based on use cases, FSMs, HTML and media objects which are developed in the phases within the waterfall oriented software development process.

1. Story specification
2. Process specification
3. Final requirement specification (use cases)
4. Dialog Structure and navigation specification (FSM)
5. Conversational data specification (static HTML, static VXML)
6. Media object specification
7. Integration of media objects and HTML
8. HTML to VXML translation
9. Test and simulation

The development process is heavily divided into phases which are based on previous ones and which produces specifications. These specifications are used for documentation purposes, too.

3.2 VoiceXML 2.0 System Architecture

Our voice system uses the free VoiceXML interpreter OptimTalk which was developed by the Laboratory of Speech and Dialogue at the Masaryk University in Brno, Czech Republic. OptimTalk supports VoiceXML 2.0 as well as DTMF input and can be added by a speech recognition component and by a SAPI 5.0 conform synthetic voice (TTS) or pre-recorded audio.

Our voice approach for information systems uses DTMF as standard input for the retrieval of information and speech recognition as a secondary input method. DTMF and speech recognition are supported by VoiceXML 2.0. The VoiceXML 2.0 based voice application could be simulated without phone hardware and it could even run on a VoiceXML 2.0 telephony system of an ASP (application service provider).

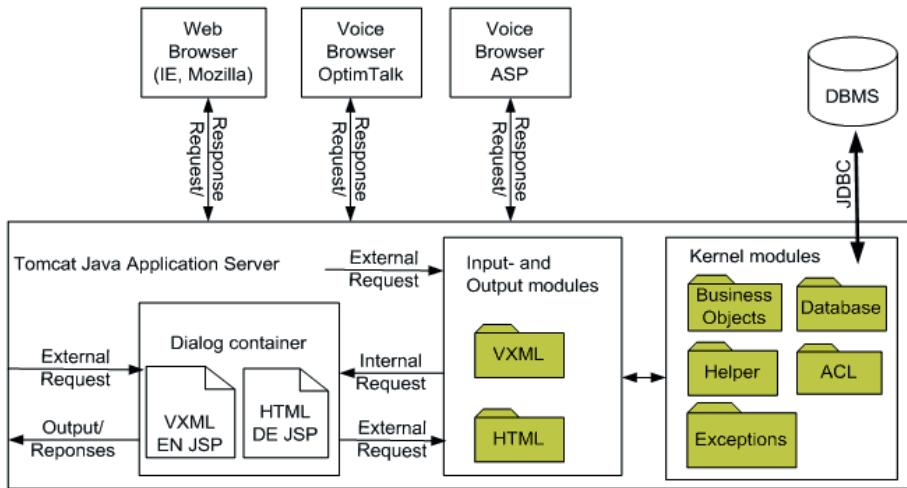


Fig. 4. VoiceXML system architecture

3.3 Future Work

We have developed a VoiceXML based speech interface for an E-Government system but there are still a lot open points which have to be analyzed.

Automated development process - Future work tends to automate this development process, especially the automatic generation of HTML pages on the basis of FSMs and the automatic translation from HTML pages to VXML pages.

Grammar Generation: dynamic and easy generation of grammars which will be used for speech recognition

Application Service Provider: VXML speech application should be tested of different telephony platforms of ASPs

Speech Recognition: VXML application should work with different speech recognition products

Speech Synthesis: VXML application should work with different speech synthesis products

Languages: VXML application with support of different languages

Integration GUI – VUI: cooperation of graphical user interface with voice user interfaces for same services and applications (E-Mail, ...)

Speech Technologies: integration of natural language technologies into a VXML system

References

1. K.-D. Schewe, B.Thalheim, Conceptual development of internet sites, Full day tutorial, ER'2000. Salt Lake City, 2000
2. Thomas Schwanzara-Benoit, Design and prototypical implementation of a natural speech interface for an E-Government system, diploma thesis at BTU Cottbus, 2004
3. B. Thalheim, Readings in fundamentals of interaction in information systems, Reprint, BTU Cottbus, 2000
4. HTML 4.01 Specification, W3C Recommendation 24 December 1999,
<http://www.w3.org/TR/html4>
5. Voice Extensible Markup Language (VoiceXML) Version 2.0, W3C Candidate Recommendation 20 February 2003, <http://www.w3.org/TR/voicexml20/>

Interpreting Semi-formal Utterances in Dialogs about Mathematical Proofs

Helmut Horacek¹ and Magdalena Wolska²

¹ Fachrichtung Informatik

Universität des Saarlandes, Postfach 15 11 50, D-66041 Saarbrücken, Germany

horacek@ags.uni-sb.de

² Fachrichtung Computerlinguistik

Universität des Saarlandes, Postfach 15 11 50, D-66041 Saarbrücken, Germany

magda@coli.uni-sb.de

Abstract. Dialogs in formal domains, such as mathematics, are characterized by a mixture of telegraphic natural language text and embedded formal expressions. Analysis methods for this kind of setting are rare and require empirical justification due to a notorious lack of data, as opposed to the richness of presentations found in genre-specific textbooks. In this paper, we focus on dedicated interpretation techniques for major phenomena observed in a recently collected corpus on tutorial dialogs in proving mathematical theorems. We combine analysis techniques for mathematical formulas and for natural language expressions, supported by knowledge about domain-relevant lexical semantics and by representations relating vague lexical to precise domain terms.

1 Introduction

Dialogs in formal domains, such as mathematics, are characterized by a mixture of telegraphic natural language text and embedded formal expressions. Acting adequately in these kinds of dialogs is specifically important for tutorial purposes since several application domains of tutorial systems are formal ones, including mathematics. Empirical findings show that flexible natural language dialog is needed to support active learning [14], and it has also been argued in favor of natural language interaction for intelligent tutoring systems [1].

To meet requirements of tutorial purposes, we aim at developing a tutoring system with flexible natural language dialog capabilities to support interactive mathematical problem solving. In order to address this task in an empirically adequate manner, we have carried out a *Wizard-of-Oz (WOz)* study on tutorial dialogs in proving mathematical theorems. In this paper, we report on interpretation techniques we have developed for major phenomena observed in this corpus. We combine analysis techniques for mathematical formulas and for natural language expressions, supported by knowledge about domain-relevant lexical semantics and by representations relating vague lexical to precise domain terms.

The outline of this paper is as follows. We first present the environment in which this work is embedded, including a description of the *WOz* experiment.

Next, we describe the link we have established between linguistic and domain knowledge sources. Then, we give details about interpretation methods for the phenomena observed in the corpus, and we illustrate them with an example. Finally, we discuss future developments.

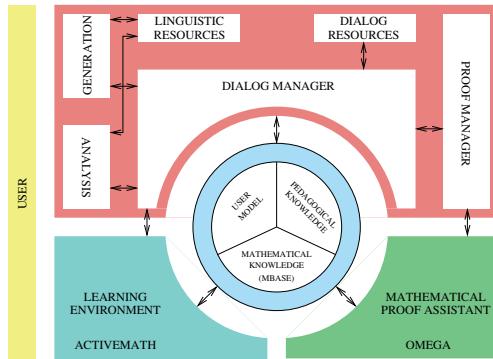


Fig. 1. DIALOG project scenario.

2 Our Project Environment

Our investigations are part of the DIALOG project¹ [5]. Its goal is (i) to empirically investigate the use of flexible natural language dialog in tutoring mathematics, and (ii) to develop an experimental prototype system gradually embodying the empirical findings. The experimental system will engage in a dialog in written natural language to help a student understand and construct mathematical proofs. In contrast to most existing tutorial systems, we envision a modular design, making use of the powerful proof system Ω MEGA [17]. This design enables detailed reasoning about the student's action and bears the potential of elaborate system responses. The scenario for the system is illustrated in Fig. 1:

- *Learning Environment*: Students take an interactive course in the relevant subfield of mathematics with the web-based system ACTIVE MATH [13].
- *Mathematical Proof Assistant (MPA)*: Checks the appropriateness of user specified inference steps wrt. to the problem-solving goal, based on Ω MEGA.
- *Proof Manager (PM)*: In the course of the tutoring session the user may explore alternative proofs. PM builds and maintains a representation of constructed proofs and communicates with the *MPA* to evaluate the appropriateness of the user's dialog contributions for the proof construction.
- *Dialog Manager*: We employ the Information-State (IS) Update approach to dialog management developed in the TRINDI project [18].

¹ The DIALOG project is part of the Collaborative Research Center on *Resource-Adaptive Cognitive Processes* (SFB 378) at University of the Saarland [15].

- *Knowledge Resources*: This includes *pedagogical knowledge* (teaching strategies), and *mathematical knowledge* (in our MBase system [11]).

We have conducted a *WOz* experiment [6] with a simulated system [7] in order to collect a corpus of tutorial dialogs in the naive set theory domain. 24 subjects with varying educational background and prior mathematical knowledge ranging from little to fair participated in the experiment. The experiment consisted of three phases: (1) *preparation and pre-test* on paper, (2) *tutoring session* mediated by a *WOz* tool, and (3) *post-test and evaluation questionnaire*, on paper again. During the session, the subjects had to prove three theorems (K and P stand for set complement and power set respectively): (i) $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$; (ii) $A \cap B \in P((A \cup C) \cap (B \cup C))$ and (iii) If $A \subseteq K(B)$, then $B \subseteq K(A)$. The interface enabled the subjects to type text and insert mathematical symbols by clicking on buttons. The subjects were instructed to enter steps of a proof rather than a complete proof at once, in order to encourage guiding a dialog with the system. The tutor-wizard's task was to respond to the student's utterances following a given algorithm [8].

3 Phenomena Observed

We have identified several kinds of phenomena, which bear some particularities of the genre and domain, and we have categorized them as follows (see Fig. 2):

- *Interleaving text with formula fragments*: Formulas may not only be introduced by natural language statements (1), they may also be enhanced by natural language function words and connectives (2), (3), or natural language and formal statements may be tightly connected (4). The latter example poses specific analysis problems, since only a part (here: variable x) of a mathematical expression (here: $x \in B$) lies within the scope of a natural language operator adjacent to it (here: negation).
- *Informal relations*: Domain relations and concepts may be described imprecisely or ambiguously using informal natural language expressions. For example, “to be *in*” can be interpreted as “element”, which is correct in (5), or as “subset”, which is correct in (6); and “both sets *together*” in (7) as “union” or “intersection”. Moreover, common descriptions applicable to collections need to be interpreted in view of the application to their mathematical counterparts, the sets: the expressions “completely outside” (8) and “completely different” (9) refer to relations on elements of the sets compared.
- *Incompleteness*: A challenge for the natural language analysis lies in the large number of unexpected synonyms, where some of them have a metonymic flavor. For example, “left side” (12) refers to a part of an equation, which is not mentioned explicitly. Moreover, the expression “inner parenthesis” (11) requires a metonymic interpretation, referring to the expression enclosed by that pair of parentheses. Similarly, the term “complement” (10) does not refer to the operator per se, but to an expression identifiable by this operator, that is, where complement is the top-level operator in the expression referred to.

	(1) Nach DeMorgan-Regel-2 ist $K((A \cup B) \cap (C \cup D)) = (K(A \cup B) \cup K(C \cup D))$ <i>According to DeMorgan-Rule-2 $K((A \cup B) \cap (C \cup D)) = (K(A \cup B) \cup K(C \cup D))$ holds</i>
Interleaving mode	(2) A auch $\subseteq K(B)$ <i>A also $\subseteq K(B)$</i>
	(3) $A \cap B$ ist \in von $C \cup (A \cap B)$, da ja $A \cap B = \emptyset$ <i>$A \cap B$ is \in of $C \cup (A \cap B)$, because $A \cap B = \emptyset$</i>
	(4) B enthält kein $x \in A$ <i>B contains no $x \in A$</i>
Informal relations	(5) Da $A \subseteq K(B)$ gilt, sind alle x , die in A sind, nicht in B <i>As $A \subseteq K(B)$ applies, all x, that are in A, are not in B</i>
	(6) $(A \cup B)$ muß in $P((A \cup C) \cap (B \cup C))$ sein, da $(A \cap B) \in (A \cap B) \cup C$ <i>$(A \cup B)$ must be in $P((A \cup C) \cap (B \cup C))$, since $(A \cap B) \in (A \cap B) \cup C$</i>
	(7) Wenn A Teilmenge von C und B Teilmenge von C dann müssen beide Mengen zusammen ebenfalls eine Teilmenge von C sein. <i>If A is a subset of C and B a subset of C, then both sets together must also be a subset of C.</i>
	(8) B muß vollständig außerhalb von A liegen, also im Komplement von A <i>B has to be entirely outside of A, so in the complement of A</i>
	(9) Dann sind A und B vollkommen verschieden, haben keine gemeinsamen Elemente <i>Then A and B are completely different, have no common elements</i>
Incompleteness	(10) $K((A \cup B) \cap (C \cup D)) = K(A \cup B) \cup K(C \cup D)$ de Morgan Regel 2 auf beide Komplemente angewendet <i>$K((A \cup B) \cap (C \cup D)) = K(A \cup B) \cup K(C \cup D)$ de Morgan rule 2 applied to both complements</i>
	(11) Distributivität von Vereinigung über Durchschnitt: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ Hier dann also: $C \cup (A \cap B) = (A \cup C) \cap (B \cup C)$ Dies für die innere Klammer <i>Distributivity of union over intersection: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ Here: $C \cup (A \cap B) = (A \cup C) \cap (B \cup C)$ This for the inner parenthesis</i>
	(12) $A \cap B$ auf der linken Seite ist \in von $C \cup (A \cap B)$, was ja nur durch C erweitert wird <i>$A \cap B$ on the left side is \in of $C \cup (A \cap B)$, which is extended only by C</i>
Operators	(13) Wenn alle A in $K(B)$ enthalten sind und dies auch umgekehrt gilt, muß es sich um zwei identische Mengen handeln <i>If all A are contained in $K(B)$ and this also holds vice-versa, these must be identical sets</i>
	(14) Mengenvereinigung ist symmetrisch <i>Set union is symmetrical</i>

Fig. 2. Examples of dialog utterances (not necessarily correct in a mathematical sense). The predicates P and K stand for power set and complement, respectively.

- *Operators*: Semantically complex operators require a domain-specific interpretation, such as “vice-versa” in (13). Occasionally, natural language referential access to mathematical concepts deviates from the proper mathematical conception. For example, the truth of some axiom, when instantiated for an operator, might be expressed as a property of that operator in natural language, such as “symmetry” as a property of “set union” (14). In the domain of mathematics, this situation is conceived as an axiom instantiation.

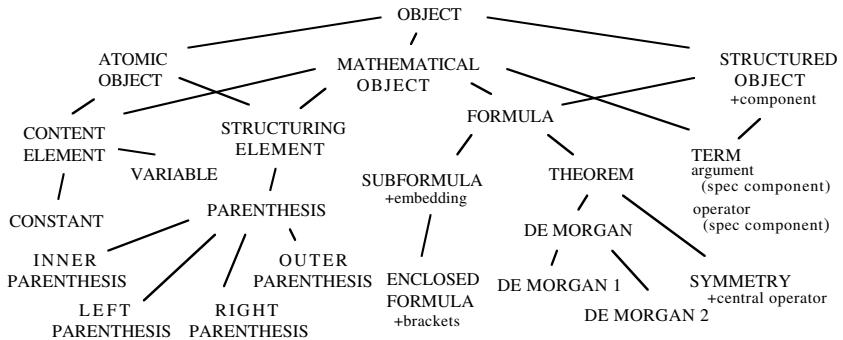


Fig. 3. A fragment of the intermediate representation of objects

4 Intermediate Knowledge Representation

In order to process adequately utterances such as the ones discussed in the previous section, natural language analysis methods require access to domain knowledge. However, this imposes serious problems, due to the fundamental representation discrepancies between knowledge bases of deduction systems, such as our system Ω MEGA, and linguistically-motivated knowledge bases, as elaborated in [10]. The contribution of the intermediate knowledge representation explained in this section is to mediate between these two complementary views.

In brief, Ω MEGA’s knowledge base is organized as an inheritance network, and representation is simply concentrated on the mathematical concepts per se. Their semantics is expressed in terms of lambda-calculus expressions which constitute precise and complete logical definitions required for proving purposes. Inheritance is merely used to percolate specifications efficiently, to avoid redundancy and to ease maintenance, but hierarchical structuring is not even imposed. Meeting communicating purposes, in contrast, does not require access to complete logical definitions, but does require several pieces of information that go beyond what is represented in Ω MEGA’s knowledge base. This includes:

- *Hierarchically organized specialization* of objects, together with their properties, and object categories for its fillers, enabling, e.g., type checking.

- The representation of *vague and general terms* which need to be interpreted in domain-specific terms in the tutorial context.
- Modeling of *typographic features* representing mathematical objects “physically”, including markers and orderings, such as argument positions.

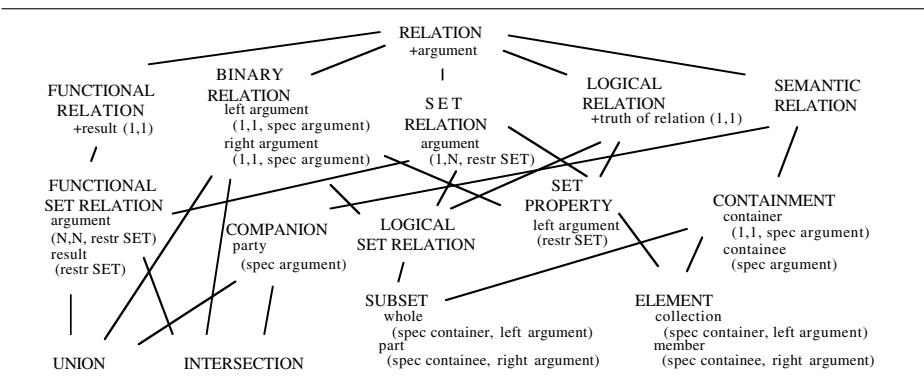


Fig. 4. A fragment of the intermediate representation of relations

In order to meet these requirements, we have built a representation that constitutes an enhanced mirror of the domain representations in Ω MEGA. It serves as an intermediate representation between the domain and linguistic models. The domain objects and relations are reorganized in a specialization hierarchy in a KL-ONE like style, and prominent aspects of their semantics are expressed as properties of these items, with constraints on the categories of their fillers. For example, the operator that appears in the definition of the symmetry axiom is re-expressed under the property *central operator*, to make it accessible to natural language references (see the lower right part of Fig. 3).

In the representation fragments in Fig. 3 and 4, objects and relations are referred to by names in capital letters, and their properties by names in small letters. Properties are inherited in an orthogonal monotonic fashion. Moreover, a specialization of a domain object may introduce further properties, indicated by a leading '+' in the property name, or it may specialize properties introduced by more general objects, which is indicated by the term 'spec' preceding the more specific property name. In addition, value restrictions on the property fillers may be specified, which is indicated by the term 'restr' preceding the filler name, and an interval enclosed in parentheses which expresses number restrictions.

These re-representations are extended in several ways. Some of the objects may be associated with procedural tests about typographical properties, which are accessible to the analysis module (not depicted in Fig. 3 and 4). They express, for instance, what makes a “parenthesis” an “inner parenthesis”, or what constitutes the “embedding” of a formula. Another important extension comprises modeling of typographic features representing mathematical objects

“physically”. This includes markers such as parentheses, as well as orderings, such as the sides of an equation. They are modeled as properties of structured objects, in addition to the structural components which make up the semantics of the logical system. Moreover, typographic properties may be expressed as parts of specializations, such as bracket-enclosed formulas as a specific kind of a (sub-)formula (Fig. 3). A further extension concerns vague and general terms, such as “containment” and “companion”, represented as semantic roles. They are conceived as generalizations of mathematical relations, in terms of the semantics associated with these roles (Fig. 4). For example, a “containment” holds between two items if the first one belongs to the second, or all its components separately do, which applies to “subset” and “element-of” relations. Similarly, “companion” comprises “union” and “intersection” operations.

5 Analysis Techniques

In this section, we present the analysis methodology and show interactions with the knowledge representation presented in Sect. 4. The analysis proceeds in 3 stages: (i) Mathematical expressions are identified, analyzed, categorized, and substituted with default lexicon entries encoded in the grammar (Sect. 5.1); (ii) Next, the input is syntactically parsed, and a representation of its linguistic meaning is constructed compositionally along with the parse (Sect. 5.2); (iii) The linguistic meaning representation is subsequently embedded within discourse context and interpreted by consulting the semantic lexicon (Sect. 5.3) and the ontology (Sect. 4).

5.1 Analyzing Formulas

The task of the mathematical expression parser is to identify mathematical content within sentences. The identified mathematical expressions are subsequently verified as to syntactic validity, and categorized as of type CONSTANT, TERM, FORMULA, 0_FORMULA (formula missing left argument), etc.

Identification of mathematical expressions within the word-tokenized text is based on simple indicators: single character tokens, mathematical symbol unicodes, and new-line characters. The tagger converts the infix notation into an expression tree from which the following information is available: surface substructure (e.g., “left side” of an expression, list of sub-expressions, list of bracketed sub-expressions), and expression type (based on the top level operator).

For example, the expression $K((A \cup B) \cap (C \cup D)) = K(A \cup B) \cup K(C \cup D)$ in utterance (10) in Fig. 2, is of type FORMULA (given the expression’s top node operator, $=$), its “left side” is the expression $K((A \cup B) \cap (C \cup D))$, the list of bracketed sub-expressions includes: $A \cup B$, $C \cup D$, $(A \cup B) \cap (C \cup D)$, etc.

5.2 Analyzing Natural Language Expressions

The task of the natural language analysis module is to produce a linguistic meaning representation of sentences and fragments that are syntactically well-

formed. The sentence meaning obtained at this stage of processing is independent of the domain-specific meaning assigned at the next stage (Sect. 5.3).

By linguistic meaning, we understand the deep semantics in the sense of the Prague School notion of sentence meaning as employed in the Functional Generative Description (FGD) [16,12]. In the Praguian FGD approach, the central frame unit of a sentence/clause is the head verb which specifies the roles of its dependents (or *participants*). Further distinction is drawn into *inner participants*, such as *Actor*, *Patient*, *Addressee*, and adverbial *free modifications*, such as *Location*, *Means*, *Direction*. To derive our set of semantic relations we generalize and simplify the collection of Praguian tectogrammatical relations in [9]. The reason for this simplification is, among others, to distinguish which of the roles have to be understood metaphorically given our specific sub-language domain (e.g., *Formula* as an *Actor*). The most commonly occurring roles in our context are those of *Cause*, *Condition*, and *Result-Conclusion*, for example *Cause* in utterance (5) and *Condition* in utterance (7) in Fig. 2. Others include *Location*, *Property*, and *GeneralRelation*.

The analysis is performed using openCCG, an open source multi-modal combinatory categorial grammar (MMCCG) parser². MMCCG is a lexicalist grammar formalism in which application of combinatory rules is controlled through context-sensitive specification of modes on slashes [2,4]. The LM, built in parallel with the syntax, is represented using Hybrid Logic Dependency Semantics (HLDS), a hybrid logic representation which allows a compositional, unification-based construction of HLDS terms with CCG [3]. Dependency relations between heads and dependents are explicitly encoded in the lexicon as modal relations.

For example, in the utterance (1) in Fig. 2 “ist” represents the meaning **hold**, and in this frame takes dependents in the tectogrammatical relations *Norm-Criterion* and *Patient*. The identified mathematical expression is categorized as of type FORMULA (reference to the structural sub-parts of the entity FORMULA are available through the information from the mathematical expression tagger). The following hybrid logic formula represents the linguistic meaning of this utterance (dMR2 denotes the lexical entry for deMorgan-Rule-2):

@h1(holds \wedge <NORM>(d1 \wedge dMR2) \wedge <PAT>(f1 \wedge FORMULA))

where h1 is the state where the proposition **holds** is true, and nominals d1 and f1 represent the dependents of kinds *Norm* and *Patient* respectively, of **holds**.

Default lexical entries (e.g. FORMULA; cf. Sect. 5.1), are encoded in the sentence parser grammar for the mathematical expression categories. At the formula parsing stage, they are substituted within the sentence in place of the symbolic expressions. The sentence parser processes the sentence without the symbolic content. The syntactic categories encoded in the grammar for the lexical entry FORMULA are *S*, *NP*, and *N*.

5.3 Domain Interpretation

The semantic lexicon defines linguistic realizations of conceptual predicates and provides a link to their domain interpretations through the ontology (cf. Sect. 4).

² <http://openccg.sourceforge.net>

Lexical semantics in combination with the knowledge encoded in the ontology allows us to obtain domain-specific interpretations of general descriptions. Moreover, productive rules for treatment of metonymic expressions are encoded through instantiation of type compatible counterparts. If more than one interpretation is plausible, no disambiguation is performed. Alternative interpretations are passed on to the Proof Manager (cf. Fig. 1) for evaluation by the theorem prover. Below we explain some of the entries the lexicon encodes (cf. Fig. 5):

(15)	$\text{contain}(\text{ACT}_{\text{type:FORMULA}}, \text{PAT}_{\text{type:FORMULA}})$	$\equiv (\text{SUBFORMULAPAT}, \text{embedding}_{\text{ACT}})$
	$\text{contain}(\text{ACT}_{\text{type:OBJECT}}, \text{PAT}_{\text{type:OBJECT}})$	$\equiv \text{CONTAINMENT}(\text{container}_{\text{ACT}}, \text{containee}_{\text{PAT}})$
(16)	$\text{in}(\text{ACT}_{\text{type:OBJECT}}, \text{LOC}_{\text{type:OBJECT}})$	$\equiv \text{CONTAINMENT}(\text{container}_{\text{LOC}}, \text{containee}_{\text{ACT}})$
(17)	$\text{outside}(\text{ACT}_{\text{type:OBJECT}}, \text{LOC}_{\text{type:OBJECT}})$	$\equiv \text{not}(\text{in}(\text{ACT}_{\text{type:OBJECT}}, \text{LOC}_{\text{type:OBJECT}}))$
(18)	$\text{common}(\text{Property}, \text{ACT}_{\text{plural(A:SET,B:SET)}})$	$\equiv \text{Property}(p_1, A) \wedge \text{Property}(p_1, B)$
(19)	$\text{common}(\text{ELEMENT}, \text{ACT}_{\text{plural(A:SET,B:SET)}})$	$\equiv \text{ELEMENT}(p_1, A) \wedge \text{ELEMENT}(p_1, B)$
(20)	$\text{different}(\text{ACT}_{\text{plural(A:SET,B:SET)}}) \equiv A \neq B$	$\equiv (e_1 \text{ ELEMENT } A \wedge e_2 \text{ ELEMENT } B \Rightarrow e_1 \neq e_2)$
	$\text{different}(\text{ACT}_{\text{plural(A:STRUCTURED OBJECT,B:STRUCTURED OBJECT)}})$	$\equiv (\text{Property}_1(p_1, A) \wedge \text{Property}_2(p_2, B) \wedge \text{Property}_1 = \text{Property}_2 \Rightarrow p_1 \neq p_2)$

Fig. 5. An excerpt of the semantic lexicon

- *Containment* The containment relation, as indicated in Fig. 4 specializes into the domain relations of (strict) SUBSET and ELEMENT. Linguistically, it can be realized, among others, with the verb “enthalten” (“contain”). The tectogrammatical frame of “enthalten” involves the roles of *Actor* (ACT) and *Patient* (PAT). Translation rules (15) serve to interpret this predicate.
- *Location* The Location relation, realized linguistically by the prepositional phrase “in... (sein)” (“be in”) involves the tectogrammatical relations of *Location* (LOC) and the *Actor* of the predicate “sein”. We consider *Location* in our domain as synonymous with Containment. Translation rule (16) serves to interpret the tectogrammatical frame of one of the instantiations of the *Location* relation. Another realization of the *Location* relation, dual to the above, occurs with the adverbial phrase “außerhalb von ... (liegen)” (“lie outside of”) and is defined as negation of Containment (17).
- *Common property* A general notion of “common property” we define as in (18). The Property here is a meta-object which can be instantiated with any relational predicate, for example, realized by a *Patient* relation as in “(A und B)_{<ACT>} haben (gemeinsame Elemente)_{<PAT>}” (“A and B have common elements”). In this case the definition (18) is instantiated as in (19).

- *Difference* The Difference relation, realized linguistically by the predicates “verschieden (sein)” (“be different”; for COLLECTION or STRUCTURED OBJECTS) and “disjunkt (sein)” (“be disjoint”; for objects of type COLLECTION) involves a plural *Actor* (e.g. coordinated noun phrases) and a *HasProperty* tectogrammatical relations. Depending on the domain type of the entity in the Actor relation, the translations are as in (20).
- *Mereological relations* Here we encode part-of relations between domain objects. These concern both physical surface and ontological properties of objects. Commonly occurring part-of relations in our domain are:

```

hasComponent(STRUCTURED OBJECTTERM,FORMULA,
              STRUCTURED OBJECTSUBTERM,SUBFORMULA)
hasComponent(STRUCTURED OBJECTTERM,FORMULA,
              STRUCTURED OBJECTENCLOSED TERM,ENCLOSED FORMULA)
hasComponent(STRUCTURED OBJECTTERM,FORMULA,
              STRUCTURED OBJECTTERM component,FORMULA component)

```

Moreover, we have from the ontology (cf. Fig. 3):

```
Property(STRUCTURED OBJECTTERM,FORMULA, componentterm side,formula side)
```

Using these definitions and polysemy rules such as polysemous(Object, Property), we can obtain interpretation of utterances such as “Dann gilt für die linke Seite, …” (“Then for the left side it holds that …”) where the predicate “gilt” normally takes two arguments of types STRUCTURED OBJECT_{TERM,FORMULA}, rather than an argument of type Property.

6 Example Analysis

Here, we present an example analysis of the utterance “B contains no $x \in A$ ” ((4) in Fig. 2) to illustrate the mechanics of the approach.

In the given utterance, the scope of negation is over a part of the formula following it, rather than the whole formula. The predicate **contain** represents the semantic relation of CONTAINMENT and is ambiguous between the domain readings of (STRICT) SUBSET, ELEMENT, and SUBFORMULA.

The formula tagger first identifies the formula $\langle x \in A \rangle$ and substitutes it with the generic entry FORMULA represented in the lexicon of the grammar. If there was no prior discourse entity for “B” to verify its type, the type is ambiguous between CONSTANT, TERM, and FORMULA. The sentence is assigned four alternative readings: “CONST contains no FORMULA”, “TERM contains no FORMULA”, “FORMULA contains no FORMULA”, and “CONST contains no CONST 0_FORMULA”. The last reading is obtained using shallow rules for modifiers (identified immediately before the formula) that take into account their possible interaction with mathematical expressions. Here, given the preceding quantifier, the expression $\langle x \in A \rangle$ has been split into its surface parts, $\langle [x] [\in A] \rangle$, [x] has been substituted with a lexical entry CONST, and [$\in A$] with an entry for a formula missing its left argument, 0_FORMULA³ (cf. Sect. 5.1). The first and the second readings are rejected

³ There are other ways of constituent partitioning of the formula to separate the operator and its arguments (they are: $\langle [x] [\in] [A] \rangle$ and $\langle [x \in] [A] \rangle$). Each of the

because of sortal incompatibility. The resulting linguistic meanings and readings of the sentence are (i) for the reading “FORMULA contains no FORMULA”:

$s:(@k1(kein \wedge <\text{RESTR}>f2 \wedge <\text{BODY}>(e1 \wedge \text{enthalten} \wedge <\text{ACT}>(f1 \wedge \text{FORMULA}) \wedge <\text{PAT}>f2)) \wedge @f2(\text{FORMULA}))$

'formula B contains no (sub-)formula 'x ∈ A'

and (ii) for the reading “CONST contains no CONST 0_FORMULA”:

$s:(@k1(kein \wedge <\text{RESTR}>x1 \wedge <\text{BODY}>(e1 \wedge \text{enthalten} \wedge <\text{ACT}>(c1 \wedge \text{CONST}) \wedge <\text{PAT}>x1)) \wedge @x1(\text{CONST} \wedge <\text{HASPROP}>(x2 \wedge 0_FORMULA)))$

'constant B contains no constant x such that x is an element of A'

The semantic lexicon is consulted to translate the readings into their domain interpretation. The relevant entries are (15) in the Fig. 5. Four interpretations of the sentence are obtained using the LMs, the semantic lexicon, and the ontology: (i) for the reading “FORMULA contains no FORMULA”:

(1) ‘it is not the case that <PAT>, formula $x \in A$, is a subformula of <ACT>, formula B ’ and, (ii) for the reading “CONST contains no CONST 0_FORMULA”

- (2a) ‘it is not the case that <PAT>, the constant x , \subseteq <ACT>, B , and $x \in A$ ’,
- (2b) ‘it is not the case that <PAT>, the constant x , \in <ACT>, B , and $x \in A$ ’,
- (2c) ‘it is not the case that <PAT>, the constant x , \subset <ACT>, B , and $x \in A$ ’.

The first interpretation, (1), is verified in the discourse context with information on structural parts of the discourse entity “B” of type FORMULA, while the other three, (2a-c), are translated into messages to the Proof Manager and passed on for evaluation in the proof context.

7 Conclusions and Future Research

In this paper, we have presented methods for analyzing telegraphic natural language text with embedded formal expressions. We are able to deal with major phenomena observed in a corpus study on tutorial dialogs about proving mathematical theorems, as carried out within the DIALOG project. Our techniques are based on an interplay of a formula interpreter and a linguistic parser which consult an enhanced domain knowledge base and a semantic lexicon.

Given the considerable demand on interpretation capabilities, as imposed by tutorial system contexts, it is hardly surprising that we are still at the beginning of our investigations. The most obvious extension for meeting tutorial purposes is the enablement to deal with errors in a cooperative manner. This requires the two analysis modules to interact in an even more interwoven way. Another extension concerns the domain-adequate interpretation of semantically complex operators such as ‘vice-versa’ as in (13) Fig. 2. ‘Vice-versa’ is ambiguous here in that it may operate on immediate dependent relations or on the embedded relations. The utterance “and this also holds vice-versa” in (13) may be interpreted

partitions obtains its appropriate type corresponding to a lexical entry available in the grammar (e.g., the $[x \in]$ chunk is of type FORMULA_0 for a formula missing its right argument). Not all the readings, however, compose to form a syntactically and semantically valid parse of the given sentence.

as “alle $K(B)$ in A enthalten sind” (“all $K(B)$ are contained in A ”) or “alle B in $K(A)$ enthalten sind” (“all B are contained in $K(A)$ ”) where the immediate dependent of the head **enthalten** and all its dependents in the *Location* relation are involved ($K(B)$), or only the dependent embedded under *GeneralRelation* (complement, K). Similarly, “head switching” operators require more complex definition. For example, the ontology defines the theorem SYMMETRY (or similarly DISTRIBUTIVITY, COMMUTATIVITY) as involving a functional operator and specifying a structural result. On the other hand, linguistically, “symmetric” is used predicatively (symmetry is predicated of a relation or function).

A further yet to be completed extension concerns modeling of actions of varying granularity that impose changes on the proof status. In the logical system, this is merely expressed as various perspectives of causality, based on the underlying proof calculus. Dealing with all these issues adequately requires the development of more elaborate knowledge sources, as well as informed best-first search strategies to master the huge search space that results from the tolerance of various kinds of errors.

References

1. V. Aleven and K. Koedinger. The Need for Tutorial Dialog to Support Self-Explanation. In *Papers from the 2000 AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications*, pages 65–73, AAAI Press, 2000.
2. J. Baldridge. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. Ph.D. Thesis, University of Edinburgh, Edinburgh, 2002.
3. J. Baldridge and G.-J. Kruijff. Coupling CCG with Hybrid Logic Dependency Semantics. In *Proc. of ACL*, pages 319–326, Philadelphia PA, 2002.
4. J. Baldridge and G.-J. Kruijff. Multi-Modal Combinatory Categorial Grammar. In *Proc. of EACL'03*, pages 211–218, Budapest, 2003.
5. C. Benzmüller, A. Fiedler, M. Gabsdil, H. Horacek, I. Kruijff-Korbayová, M. Pinkal, J. Siekmann, D. Tsovaltzi, B. Vo, and M. Wolska. Tutorial Dialogs on Mathematical Proofs. In *IJCAI Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems*, pages 12–22, 2003.
6. C. Benzmüller, A. Fiedler, M. Gabsdil, H. Horacek, I. Kruijff-Korbayová, M. Pinkal, J. Siekmann, D. Tsovaltzi, B. Vo, and M. Wolska. A Wizard-of-Oz Experiment for Tutorial Dialogues in Mathematics. In *AIED2003 — Supplementary Proceedings of the 11th International Conference on Artificial Intelligence in Education*, pages 471–481, Sidney, Australia, 2003.
7. A. Fiedler and M. Gabsdil. Supporting Progressive Refinement of Wizard-of-Oz Experiments. In *Proc. of the ITS 2002 – Workshop on Empirical Methods for Tutorial Dialogue*, pages 62–69, San Sebastian, Spain, 2002.
8. A. Fiedler and D. Tsovaltzi. Automating Hinting in Mathematical Tutorial Dialogue. In *Proc. of the EACL-03 Workshop on Dialogue Systems: Interaction, Adaptation and Styles of Management*, pages 45–52, Budapest, 2003.
9. E. Hajíčová, J. Panevová, and P. Sgall. A Manual for Tectogrammatical Tagging of the Prague Dependency Treebank. TR-2000-09, Charles University, Prague, 2000.
10. H. Horacek, A. Fiedler, A. Franke, M. Moschner, M. Pollet, and V. Sorge. Representation of Mathematical Objects for Inferencing and for Presentation Purposes. In *Proc. of EMCSR-2004*, pages 683–688, Vienna, 2004.

11. M. Kohlhase and A. Franke. MBase: Representing Knowledge and Context for the Integration of Mathematical Software Systems. *Journal of Symbolic Computation*, 32(4):365–402, 2000.
12. G.-J. Kruijff. *A Categorial-Modal Logical Architecture of Informativity: Dependency Grammar Logic & Information Structure*. Ph.d. Dissertation, Charles University, Prague, 2001.
13. E. Melis et al. ACTIVEMATH: A Generic and Adaptive Web-Based Learning Environment. *Artificial Intelligence in Education*, 12(4):385-407, 2001.
14. J. Moore. What Makes Human Explanations Effective? In *Proc. of the Fifteenth Annual Conference of the Cognitive Science Society*, Hillsdale, NJ. Earlbaum, 1993.
15. SFB 378 web-site: <http://www.coli.uni-sb.de/sfb378/>.
16. P. Sgall, E. Hajičová, and J. Panevová. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Reidel Publishing Company, Dordrecht, 1986.
17. J. Siekmann et al. Proof Development with Ω MEGA. In *Proceedings of the 18th Conference on Automated Deduction*, pages 144–149, Copenhagen, Denmark, 2002.
18. TRINDI project: <http://www.ling.gu.se/research/projects/trindi/>.

Event Ordering Using TERSEO System*

Estela Saquete, Rafael Muñoz, and Patricio Martínez-Barco

Grupo de investigación del Procesamiento del Lenguaje y Sistemas de Información.

Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante.
Alicante, Spain

{stela,rafael,patricio}@dlsi.ua.es

Abstract. In this paper, a method of event ordering based on temporal information resolution is presented. This method consists of two main steps: on the one hand, the recognition and resolution of the temporal expressions that can be transformed on a date, and therefore these dates establish an order between the events that contain them. On the other hand, the detection of temporal signals, for example *after*, that can not be transformed on a concrete date but relate two events in a chronological way. This event ordering method can be applied to Natural Language Processing systems like for example: Summarization, Question Answering, etc.

1 Introduction

Nowadays, the information society needs a set of tools for the increasing amount of digital information stored in the Internet. Documental database applications help us to manage this information. However, documental database building requires the application of automatic processes in order to extract relevant information from texts.

One of these automatic processes is event ordering by means of temporal information. Usually, a user needs to obtain all the information related to a specific event. To do this, he must know the relationships between other events, and their chronological information. The automatic identification of temporal expressions associated with events, temporal signals that relate events, and further treatments of them, allow the building of their chronographic diagram. Temporal expressions treatment is based on establishing relationships between concrete dates or time expressions (25th December 2002) and relative dates or time expressions (the day before). Temporal signals treatment is based on determining the temporal relationship between the two events that the signal is relating. Using all this information, the application of event-ordering techniques allows us to obtain the desired event ordering.

This paper has been structured in the following way: first of all, section 2 shows a short introduction to the main contributions of previous work. Then, section 3 describes the Event Ordering system and the different units of the system:

* This paper has been supported by the Spanish government, projects FIT-150500-2002-244, FIT-150500-2002-416, TIC-2003-07158-C04-01 and TIC2000-0664-C02-02

the Temporal Information Detection unit, the Temporal Expression Coreference Resolution unit, the Ordering Keys unit and the Event Ordering unit. Following this, there is a graphical example of how the event ordering method works. In section 4, the application of this event ordering method in one task of NLP (Question Answering) is explained. Finally, the evaluation of TERSEO system in Spanish and some conclusions are shown.

2 Previous Work

At the moment there are different kinds of systems that cope with the event ordering issue. Some of the current systems are based on knowledge like Filatova and Hovy[3] which describes a procedure for arranging into a time-line the contents of news stories describing the development of some situation. The system is divided in two main parts: firstly, breaking sentences into event-clauses and secondly resolving both explicit and implicit temporal references. Evaluations show a performance of 52%, compared to humans. Schilder and Habel[10] system is knowledge based as well. This system detects temporal expressions and events and establishes temporal relationships between the events. Works like Mani and Wilson[5] develop an event-ordering component that aligns events on a calendric line, using tagged TIME expressions. By contrast, to some other important systems are based on Machine Learning and focused on Event Ordering, for instance, Katz and Arosio[4], Setzer and Gaizauskas[11]. This last one is focused on annotating Event-Event Temporal Relations in text, using a time-event graph which is more complete but costly and error-prone.

Although systems based on Machine Learning obtain high precision results applied to concrete domains, these results are lower when these kind of systems are applied to other domains. Besides, they need large annotated corpus. On the other hand, systems based on knowledge have a greater flexibility to be applied to any domain. Our proposal is a hybrid system (TERSEO) that takes profit of the advantages of both kind of systems. TERSEO has a knowledge database but this database has been extended using an automatic acquisition of new rules for other languages[7]. That is why TERSEO is able to work in a multilingual level. However, in this article we are focused on the event ordering method based on TERSEO system. The description of the Event Ordering System is made in the following section.

3 Description of the Event Ordering System

The graphic representation of the system proposed for event ordering is shown in Figure 1. Temporal information is detected in two steps. First of all, temporal expressions are obtained by the Temporal Expression Detection Unit. After that, the Temporal Signal Detection Unit returns all the temporal signals. The Temporal Expressions (TEs) that have been recognized are introduced into the resolution unit, which will update the value of the reference (document's date at first) according to the date it refers to and generates XML tags for each

expression. These tags are part of the input of an event ordering unit. The temporal signals that have been obtained are introduced into a unit that obtains the ordering keys for each temporal signal. The ordering key establishes the order between two events and is used by the event ordering unit as well. With all this information the event ordering unit is able to return the ordered text. Temporal Expressions and Temporal signals are explained in detail below.

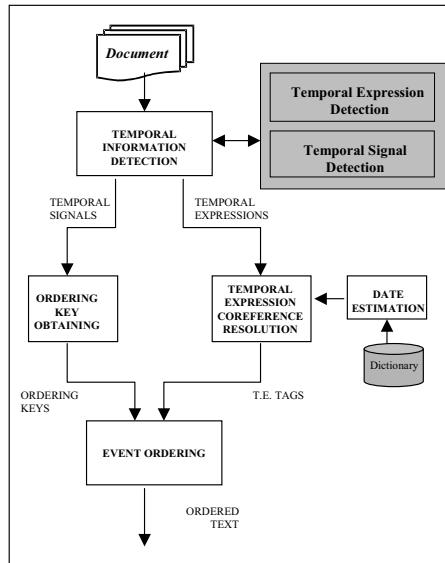


Fig. 1. Graphic representation of the Event Ordering System

3.1 Temporal Information Detection

The Temporal Information Detection Unit is divided in two main steps:

- Temporal Expressions Detection
- Temporal Signal Detection

Both steps are fully explained in following sections but they share a common preprocessing of the texts. Texts are tagged with lexical and morphological information by a Pos Tagger and this information is the input to a temporal parser. This temporal parser is implemented using an ascending technique (chart parser) and it is based on a temporal grammar [9].

Temporal Expressions Detection. One of the main tasks involved in trying to recognize and resolve temporal expressions is to classify them, because the way of solving them depends on the type of expression. In this paper, two proposals

for the classification of temporal expressions are shown. The first classification is based on the kind of reference. This classification is focused on recognizing the kind of expression when this enters the system and needs to be resolved. In addition, another type of classification is presented. This one is focused on the kind of output returned by the system for that type of expression.

– **Classification of the expression based on the kind of reference**

• **Explicit Temporal Expressions.**

- * *Complete Dates with or without time expressions:* “11/01/2002” (*01/11/2002*), “el 4 de enero de 2002” (*January 4th, 2002*),...
- * *Dates of Events:*

- Noun Phrase with explicit date: “el curso 2002-2003” (*2002-2003 course*). In this expression, “course” denotes an event
- Noun Phrase with a well-known date: “Navidad” (*Christmas*),...

• **Implicit Temporal Expressions.**

- * Expressions that refer to the *Document date*:

- Adverbs or adverbial phrases: “ayer” (*yesterday*),...
- Noun phrases: “el próximo mes” (*the next month*),...
- Prepositional phrases: “en el mes pasado” (*in the last month*),...

- * Expressions that refers to *another date*:

- Adverbial Phrases: “durante el curso” (*during the course*),...
- Noun Phrases: “un mes después” (*a month later*), “después de la próxima Navidad” (*after next Christmas*),... For example, with the expression “after next Christmas” it is necessary to resolve the TE “next Christmas” and then apply the changes that the word “after” makes on the date obtained.
- Prepositional Phrases: “desde Navidad” (*from Christmas*), “desde la anterior Navidad” (*since last Christmas*),...

– **Classification by the representation of the temporal value of the expression**

• **Concrete.** All those that give back a concrete day or/and time with format: dd/mm/yyyy (hh:mm:ss) (mm/dd/yyyy (hh:mm:ss)), for example: “ayer” (*yesterday*).

• **Period.** All those expressions that give back a time interval or range of dates: [dd/mm/yyyy-dd/mm/yyyy] ([mm/dd/yyyy-mm/dd/yyyy]), for example: “durante los siguientes cinco días” (*during the five following days*).

• **Fuzzy.** It gives back an approximate time interval because it does not know the concrete date that the expression refers to. There are two types:

- * *Fuzzy concrete.* If the given result is an interval but the expression refers to a concrete day within that interval, and we do not know it accurate. For that reason we must give back the approach of the interval, for example: “un día de la semana pasada” (*a day of the last week*),...

- * *Fuzzy period.* If the expression refers to an interval contained within the given interval, for instance: “hace unos días” (*some days before*), “durante semanas” (*during weeks*),...

In section 5, we will see how the system is able to solve great part of these temporal expressions, that have been recognized by the Temporal Expression Unit.

Temporal Signals Detection. The temporal signals relate the different events in texts and establish a chronological order between these events. In an experimental way, after the study of a training corpus, a set of temporal signals has been obtained, and some of them are emphasized here: *después (after)*, *cuando (when)*, *antes (before)*, *durante (during)*, *previamente (previously)*, *desde ... hasta... (from ... to ...)*, *en (on, in)*, *mientras (while)*, *por (for)*, *en el momento de (at the time of)*, *desde (since)*,etc.

3.2 Temporal Expression Coreference Resolution

Temporal Expression Coreference Resolution is organized in two different tasks:

- Anaphoric relation resolution based on a temporal model
- Tagging of temporal Expressions

Every task is explained next.

Anaphoric relation resolution based on a temporal model. For the anaphoric relation resolution we use an inference engine that interprets every reference named before. In some cases the references are estimated using the document's date (FechaP). Others refer to a date named before in the text that is being analyzed (FechaA). For these cases, a temporal model that allows to know on what date the dictionary operations are going to be done, is defined. This model is based on the two rules below and it is only applicable to these dates that are not FechaP, since for FechaP there is nothing to resolve:

1. By default, the newspaper's date is used as a base referent (TE) if it exists, if not, the system date is used.
2. If a non-anaphoric TE is found, this is stored as FechaA. This value is updated every time that a non-anaphoric TE appears in the text.

In Table 1¹ some of the entries in the dictionary used in the inference engine are shown. The unit that makes the estimation of the dates will accede to the right entry in the dictionary in each case and it will apply the function specified obtaining a date in the format dd/mm/yyyy (*mm/dd/yyyy*) or a range of dates. So, at that point the anaphoric relation will have been resolved.

¹ The operation '+1' in the dictionary is able to interpret the dates in order to give back a valid date. For example, if the Month (date) function gives back 12 and the operation '+1' is done on that value, the given back value will be 01, increasing a year.

Table 1. Sample of some of the entries in the dictionary

REFERENCE	DICCIONARY ENTRY
‘ayer’ (<i>yesterday</i>)	Day(FechaP)-1/Month(FechaP)/Year(FechaP)
‘mañana’ (<i>tomorrow</i>)	Day(FechaP)+1/Month(FechaP)/Year(FechaP)
‘durante el mes siguiente’ (<i>during the following month</i>)	[DayI/Month(FechaA)+1/Year(FechaA)-- DayF/Month(FechaA)+1/Year(FechaA)]
num+‘años siguientes’ (<i>num years later</i>)	[01/01/Year(FechaA)+num -- 31/12/Year(FechaA)+num]
‘un día antes’ (<i>a day before</i>)	Day(FechaA)-1/Month(FechaA)/Year(FechaA)
‘días después’ (<i>some days later</i>)	>>>FechaA
‘días antes’ (<i>some days before</i>)	<<<FechaA

Tagging of temporal expressions. Several proposals for the annotation of TEs have arisen in the last few years Wilson et al.[13], Katz and Arosio [4], TIMEML[6], etc. since some research institutions have started to work on different aspects of temporal information. In this section, our own set of XML tags is defined in order to standardize the different kinds of TEs. We have defined a simple set of tags and attributes that adjust to the necessities of our system, without complicating it. Besides, they could be transformed to other existing formats, like TIMEML, at any time. These tags show the following structure:

- For Explicit Dates:

```
<DATE_TIME ID="“value”” TYPE="“value”” VALDATE1="“value””  
VALTIME1="“value”” VALDATE2="“value”” VALTIME2="“value””  
VALORDER="“value””>expression</DATE_TIME>
```

- For Implicit Dates:

```
<DATE_TIME_REF ID="“value”” TYPE="“value”” VALDATE1="“value””  
VALTIME1="“value”” VALDATE2="“value”” VALTIME2="“value””  
VALORDER="“value””>expression</DATE_TIME_REF>
```

DATE_TIME is the name of the tag for explicit TEs and DATE_TIME_REF is the name of the tag for implicit TEs. Every expression has an numeric ID to be identified and VALDATE# and VALTIME# store the range of dates and times obtained from the inference engine, where VALDATE2 and VALTIME2 is only used to establish ranges. Also, VALTIME1 could be omitted if only a date is specified. VALDATE2, VALTIME1 and VALTIME2 are optional args. VALORDER is the attribute where the event ordering unit will specify the ordering value, at first there is no value for this attribute. After that, a structured document is obtained. The use of XML allows us to take advantage of the XML schema in which the tag language is defined. This schema lets an application know if the XML file is valid and well-formed. A parser of our XML needs to be defined to make the information useful.

3.3 Ordering Keys Obtaining

The temporal signals obtained by the Temporal Signal Detection Unit are used by this unit to obtain the ordering keys. The study of the corpus revealed a set of temporal signals. Each temporal signal denotes a relationship between the dates of the events that it is relating. For example, in EV1 S EV2, the signal S denotes a relationship between EV1 and EV2. Assuming that F1 is the date related to the first event and F2 is the date related to the second event, the signal will establish a certain order between these events. This order will be established by the ordering unit. Some of the ordering keys, which are the output of this unit, are shown in Table 2.

Table 2. Output of the ordering Key Obtaining Unit

SIGNAL	ORDERING KEY
After	$F1 > F2$
When	$F1 = F2$
Before	$F1 < F2$
During	$F2i \leq F1 \leq F2f$
Previously	$F1 > F2$
From F2 to F3	$F2 \leq F1 \leq F3$
About F2 -- F3	$F2 \leq F1 \leq F3$
On / in	$F1 = F2$
While	$F2i \leq F1 \leq F2f$
For	$F2i \leq F1 \leq F2f$
At the time of	$F1 = F2$
Since	$F1 > F2$

3.4 Event Ordering Method

Event ordering in natural language written texts is not a trivial task. Firstly, a process to identify events must be done. Then, the relationship between the events or between the event and the date when the event occurs must be identified. Finally, the ordering of events must be determined according to their temporal information. This temporal information could be dates, temporal expressions or temporal signals. We have trivialized the task of identifying events. We only will identify events as the sentence that includes some kind of TE or a sentence that is related to another sentence by a temporal signal.

Using the XML tags and the ordering keys, the event ordering module runs over the text building a table that specify the order and the date, if there is any, of every event. The order is established according to the following rules:

1. EV1 is previous to EV2:
 - if the range of VALDATE1, VALTIME1, VALDATE2, VALTIME2 associated with EV1 is prior to and not overlapping the range associated with EV2.

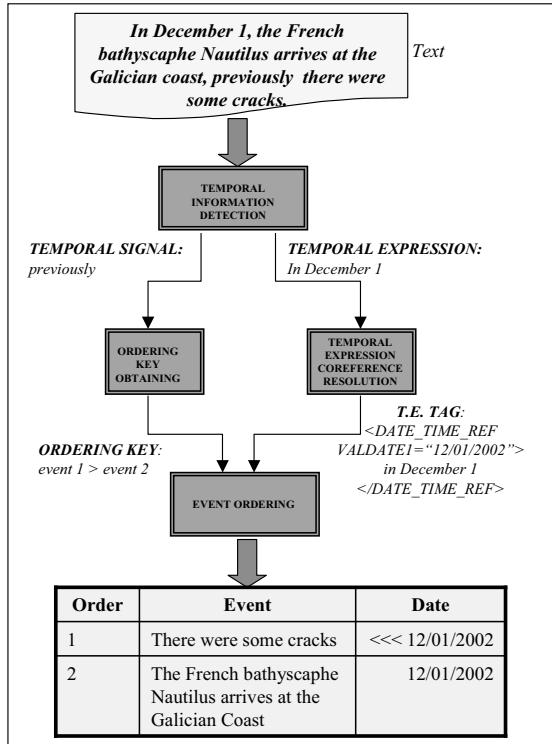


Fig. 2. Graphical example of Event Ordering

- or, if the ordering key that relate both events is:

$EV1 < EV2$

2. $EV1$ is concurrent to $EV2$:

- if the range of $VALDATE1$, $VALTIME1$, $VALDATE2$, $VALTIME2$ associated with $EV1$ overlaps the range associated with $EV2$.
- or, if the ordering key that relate both events is:

$EV1 = EV2$ or $EV1i \leq EV2 \leq EV1f$

The system will assign a sequential order number to every event in the table, having the same order number for concurrent events.

In Figure 2 an example is shown. Newspaper's date:30/12/2002

4 Application of Event Ordering in NLP Tasks

Event Ordering can be applied in different tasks in the field of Natural Language Processing. Some of the applications in which event ordering is useful are for example: Summarization, Question Answering, etc. In particular, we have developed a method to apply Event Ordering to Temporal Question Answering[8].

Temporal Question Answering is not a trivial task due to the complexity that temporal questions can achieve. Current Question Answering systems can deal with simple questions requiring a date as answer or questions that use explicit temporal expressions in their formulation. Nevertheless, more complex questions referring to the temporal properties of the entities being questioned and their relative temporal ordering in the question are beyond the scope of current Question Answering systems. These complex questions consist of two or more events, related with a temporal signal, which establishes the order between the events in the question. This situation allows us to divide, using Temporal Signals, the complex question into simple ones, which the current Question Answering systems are able to resolve, and recompose the answer using the ordering key to obtain a final answer to the complex question.

An example of the how the system works with the question: *Where did Bill Clinton study before going to Oxford University?* is shown here:

1. First of all, the unit recognizes the temporal signal, which in this case is *before*.
2. Secondly, the complex question is divided into simple ones.
 - Q1: Where did Bill Clinton study?
 - Q2: When did Bill Clinton go to Oxford University?
3. A general purpose Question Answering system answers the simple questions, obtaining the following results:
 - Answer for Question 1: Georgetown University (1964-1968)
 - Answer for Question 1: Oxford University (1968-1970)
 - Answer for Question 1: Yale Law School (1970-1973)
 - Answer for Question 2: 1968
4. All those answers that do not fulfill with the constraint established by the ordering key are rejected.
5. After that, the final answer to complex question is *Georgetown University*.

5 System Evaluation

In order to carry out an evaluation of this system, a manual annotation of texts has been made by two annotators with the purpose of comparing it with the automatic annotation that produces the system. For that reason, it is necessary to confirm that the manual information is trustworthy and it does not alter the results of the experiment. Carletta [2] explains that to assure a good annotation is necessary to make a series of direct measurements that are: stability, reproducibility and precision, but in addition to these measurements the reliability must measure the amount of noise in the information. The authors argue that, due to the amount of agreement by chance that can be expected depends on the number of relative frequencies of the categories under test, the reliability for the classifications of categories would have to be measure using the factor *kappa* defined in Siegel and Castellan [12]. The factor *kappa* (*k*) measures the affinity in agreement between a set of annotator when they make categories judgments.

In our case, there is only one class of objects and there are three objects within this class: objects that refer to the date of the article, objects which refer to the previous date and objects that refer to another date different from the previous ones.

After carrying out the calculation, a value $k=0.953$ was obtained. According to the work of Carletta [2], a measurement of k like $0.68 < k < 0.8$ means that the conclusions are favorable, and if $k > 0.8$ means total reliability exists between the results of both annotators. Since our value of k is greater than 0.8, it is guaranteed that a total reliability in the conducted annotation exists and therefore, the results of obtained precision and recall are guaranteed.

In order to evaluate the event ordering method, an evaluation of the TERSEO system in a monolingual level (Spanish) was carried out. The establishment of a correct order between the events implies that the resolution is correct and the events are placed on a timeline, as it is shown in Figure 3. For this reason, we have made an evaluation of the resolution of the Temporal Expressions.

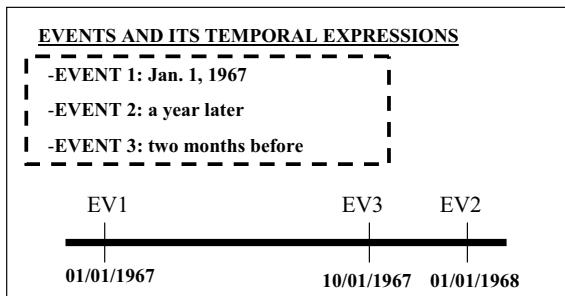


Fig. 3. Event Ordering based on TE resolution

Two corpora formed by newspaper articles in Spanish were used. The first set has been used for training and it consists of 50 articles. Thus, after making the opportune adjustments to the system, the optimal results of precision and recall obtained are in the table 3:

Although the obtained results are highly successful, we have detected some failures that have been deeply analyzed. As can be observed in the results, our system could be improved in some aspects. Below, a study of the problems detected and their possible improvements are shown:

- In the newspaper's articles, sometimes there are expressions like “*el sábado hubo cinco accidentes*” (Saturday there were five accidents). To resolve these kind of references we need context information of the sentence where the reference is. That information could be the time of the sentence's verb. If the verb is a past verb, it indicates that it is necessary to solve a reference like “*el sábado pasado*” (last Saturday), whereas if it is a future verb it refers to “*el sábado próximo*” (the next Saturday). Because our system does not

Table 3. Evaluation of the system

	TRAINING	TEST
No Art.	50	50
Real Ref	238	199
Treated Ref.	201	156
Successes	170	138
Precision	84%	91%
Recall	71%	73%
Coverage	84%	80%

use semantic or context information we assume this kind of reference refers to the last day, not the next, because the news usually tells us facts which occurred previously.

- Our system is not able to resolve temporal expressions that contain a well-known event, for instance: “*two days before the war...*”. In order to solve this kind of expressions, some extra knowledge of the world is necessary, and we are not able to access this kind of information nowadays.

6 Conclusions

In this article a method of event ordering has been presented. This method is based on the detection of keywords (temporal signals) and the resolution of the temporal information associated to the event. This method can be applied to multilingual texts because TERSEO system solves the temporal expressions in this type of texts. The obtained results show that this tool can be used to improve other systems of NLP as for example: Question Answering systems, with questions of temporal information or Summarization systems. Nowadays, an application of this work is being used applied to a Question Answering system[8].

As future work, two tasks will be considered:

- The system will cope with the resolution of temporal expressions considering context information or world knowledge.
- An evaluation of TERSEO system in a multilingual level is being prepared.

References

1. ACL, editor. *Proceedings of the 2001 ACL-EACL, Workshop on Temporal and Spatial Information Processing*, Toulouse, France, 2001.
2. J. Carletta and et al. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23(1):13–32, 1997.
3. E. Filatova and E. Hovy. Assigning time-stamps to event-clauses. In ACL [1], pages 88–95.
4. G. Katz and F. Arosio. The annotation of temporal information in natural language sentences. In ACL [1], pages 104–111.

5. I. Mani and G. Wilson. Robust temporal processing of news. In ACL, editor, *Proceedings of the 38th Meeting of the Association of Computational Linguistics (ACL 2000)*, Hong Kong, October 2000.
6. D. Radev and B. Sundheim. Using timeml in question answering. <http://www.cs.brandeis.edu/~jamesp/arda/time/documentation/TimeML-use-in-qa-v1.0.pdf>, 2002.
7. E. Saquete, P. Martínez-Barco, and R. Muñoz. Automatic multilinguality for time expression resolution. In MICAI, editor, *Proceedings Mexican International Conference on Artificial Intelligence*, Mexico D.F., Mexico, April 2004.
8. E. Saquete, P. Martínez-Barco, R. Muñoz, and J.L. Vicedo. Decomposition of complex temporal questions for question answering systems. Technical report, DLSI, 2004.
9. E. Saquete, R. Muñoz, and P. Martínez-Barco. Terseo: Temporal expression resolution system applied to event ordering. In TSD, editor, *Proceedings of the 6th International Conference ,TSD 2003, Text, Speech and Dialogue*, pages 220–228, Ceske Budejovice,Czech Republic, September 2003.
10. F. Schilder and C. Habel. From temporal expressions to temporal information: Semantic tagging of news messages. In ACL [1], pages 65–72.
11. A. Setzer and R. Gaizauskas. On the importance of annotating event-event temporal relations in text. In LREC, editor, *Proceedings of the LREC Workshop on Temporal Annotation Standards, 2002*, pages 52–60, Las Palmas de Gran Canaria,Spain, 2002.
12. S. Siegel and J. Castellan. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, 2nd edition, 1988.
13. G. Wilson, I. Mani, B. Sundheim, and L. Ferro. A multilingual approach to annotating and extracting temporal information. In ACL [1], pages 81–87.

The Role of User Profiles in Context-Aware Query Processing for the Semantic Web

Veda C. Storey¹, Vijayan Sugumaran², and Andrew Burton-Jones¹

¹ J. Mack Robinson College of Business

Georgia State University, Box 4015

Atlanta, GA 30302

{vstorey, abjones}@gsu.edu

² School of Business Administration

Oakland University

Rochester, MI 48309

sugumara@oakland.edu

Abstract. Many queries processed on the World Wide Web do not return the desired results because they fail to take into account the context of the query and information about user's situation and preferences. In this research, we propose the use of user profiles as a way to increase the accuracy of web pages returned from the Web. A methodology for creating, representing, and using user profiles is proposed. A frame-based representation captures the initial user's profile. The user's query history and post query analysis is intended to further update and augment the user's profile. The effectiveness of the approach for creating and using user profiles is demonstrated by testing various queries.

1 Introduction

The continued growth of the World Wide Web has made the retrieval of relevant information for a user's query difficult. Search engines often return a large number of results when only a few are desired. Alternatively, they may come up "empty" because a small piece of information is missing. Most search engines perform on a syntactic basis, and cannot assess the usefulness of a query as a human would who understands his or her own preferences and has common sense knowledge of the real world. The problems associated with query processing, in essence, arise because context is missing from the specification of a query. The Semantic Web has been proposed to resolve context problems by documenting and using semantics [1]. This research investigates how user profiles can be used to improve web searches by incorporating context during query processing. The objectives are to: 1) develop a heuristics-based methodology to capture, represent, and use user profiles; 2) incorporate the methodology into a prototype, and 3) test the effectiveness of the methodology. The contribution of the research is to help realize the Semantic Web by capturing and using the semantics of a query through user profiling. Our research contributes, not through a new approach to building, representing, or using a user profile, but by show

ing how user profiles can be used as part of an integrated approach that utilizes lexical, ontological, and personal knowledge.

2 Related Research

Information retrieval involves four steps: 1) understanding the task, 2) identifying the information need, 3) creating the query, and 4) executing the query [2]. This research focuses on the third step, namely, query creation.

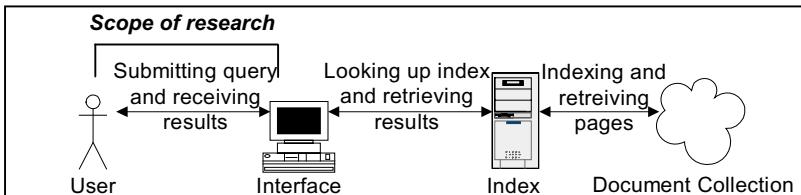


Fig. 1. High-level Architecture and Process of a Web Search

In Figure 1 a user sends a query to a search engine using keywords and syntactical operations. The search engine is a black box: a user submits a query and the search engine lists the results. This search engine remains a black box in this research as it focuses on refining the query before it reaches the interface. Refinement is important because users submit short queries that return many irrelevant results [12]. A major challenge is *contextual* retrieval: combining search technologies and contextual knowledge to provide the most appropriate answer. One obstacle to contextual retrieval is the lack of intelligence in Web-search systems. The Semantic Web is important in achieving contextual retrieval because terms on Web pages will be marked up using ontologies that define each term's meaning. It will be a long time, however, before markup on pages becomes the norm. Thus, this research provides a methodology for using contextual knowledge to improve query refinement to increase the relevance of results on the current and Semantic Web.

Context is: (1) the parts of a discourse that surround a word/passage and can shed light on its meaning, and (2) the interrelated conditions in which something exists (Merriam-Webster). For web queries, the first definition refers to a query's lexical context; the second to the user submitting the query. By considering both, we aim to achieve a fully contextualized query. A query is relevant if it satisfies a user's information need (Borlund, 2003). Thus, a *contextualized* query returns relevant results by accounting for (a) the *meaning* of query terms and (b) the user's *preferences*. An optimal contextual query will minimize the distance between the information need, I, and the query, Q. Distance ($I \rightarrow Q$) is minimized by $\text{Min} (D_c, D_p, D_L)$, where:

- D_c = use of the wrong concepts in the query to represent the information need
- D_p = lack of preferences in the query to constrain the concepts requested
- D_L = lack of precision in the language used in the query terms

Why would a user write a query with positive levels of D_c , D_p , D_L ? We postulate:

- Postulate 1: D_c and D_p will be higher when a user lacks domain knowledge.

- Postulate 2: D_p will be higher when a user lacks knowledge of preferences.
- Postulate P3: D_L will be higher when a user lacks lexical knowledge.

Prior research suggests three techniques for minimizing D_C , D_p , D_L :

- Ontologies: Ontologies consist of terms, their definitions, and axioms relating them [7]. They can minimize D_C by helping users understand relationship between concepts. For example, to find a job as a Professor, an ontology might suggest relevant related terms, such as teaching and research.
- Lexicons: Lexicons comprise the general vocabulary of a language. Lexicons can be used to minimize D_L by identifying a term's meaning by its connection to other terms and by selecting terms with minimal ambiguity. For example, in finding out about being a Professor, a lexicon might suggest that the user avoids using terms such as 'Chair' that have multiple meanings.
- User Profiles: User profiles are a way of stating preferences about a concept. They can minimize D_p by serving as a constraint on the range of instances that will be retrieved by the query. For example, a profile could limit one's query for Professorial jobs to positions in the United Kingdom.

Our prior research developed a Semantic Retrieval System (SRS) [3] (Figure 2) that uses lexical sources from WordNet and ontological sources from the DARPA ontology library to parse natural language queries and create contextualized queries. Given a set of terms, the system expands the query using lexically and domain-related terms to contextualize the query. The query terms form root nodes in a semantic network that expands by adding related terms and shrinks by removing terms of unwanted context, iterating towards an effective query. The system does not include user profiles, so it can not create fully contextualized queries. This research attempts to achieve this by capturing users' preferences via a user profile module.

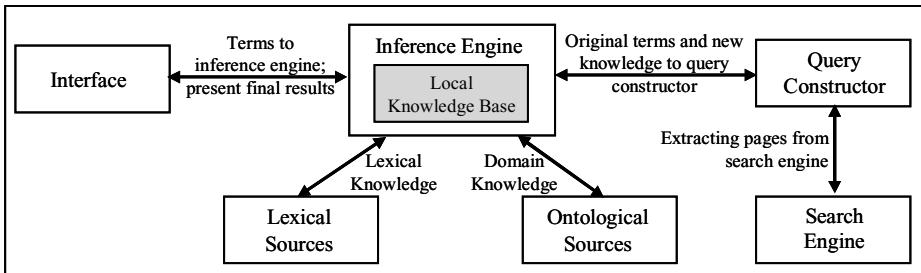


Fig. 2. Query Expansion and Semantic Web Retrieval Architecture

3 Integrating User Profiles into Query Contextualization

Consider the query in Table 1. To retrieve relevant results, much personal information is needed; merely parsing for key words will be not be sufficient. Given that most users have preferences when querying (per Table 1), it may be unclear why a user would exclude them from a query. Postulate 2 suggests two reasons:

Table 1. Sample Query (adapted from [1])

Natural Language Query	Personal Information Needed
Mom needs to have a series of physical therapy sessions. Biweekly or something.... Set up the appointments.	Prescribed treatment, Preferred providers, Insurance plan, Preferred distance from home, Preferred quality of service, Preferred appointment times.

1. *Lack of domain knowledge*: the user lacks knowledge of the domain and, thus, about possible preferences (e.g., the user does not know that physiotherapists are covered by insurance, and so does not include her/his insurance preferences).

2. *Delegated query*: the user (or agent) is submitting a query on behalf of another user (e.g., ‘Mom’ in Table 1) and does not know that user’s preferences.

Considering postulates 1-3 together yields a decision matrix (Table 2).

Table 2. Applicability of Contextual Knowledge Source for Search Query*

		Do Query Yourself	Delegated Query
		1	2
Domain Expert	Do not use ontology	1	Do not use ontology
	Use lexicon if uncertain/ambiguous		Use lexicon if uncertain/ambiguous
	Do not use personal profile		Use user profile (personal)
Domain Novice	Use ontology	3	Use ontology
	Use lexicon if uncertain/ambiguous		Use lexicon if uncertain/ambiguous
	Use user profile (stereotype)		Use user profile (stereotype & personal)

* For simplicity, lexical knowledge is not shown as a third dimension

The Semantic Retrieval System considers the conditions in Table 2 to determine what contextual knowledge to invoke. A user can manually select a level for each variable (domain expertise, query writer, and language certainty) or the system can infer a user’s level. The heuristics used for inferences are:

- *Domain expertise*: A user is an expert if he or she submits the same query as another user listed in the system’s history as an expert.
- *Query writer*: A user is the previous user if s/he submits a query that contains a term that is the same or one degree separated from a term in the prior query.
- *Language uncertainty*: A user requires a lexicon if he or she submits a term that has a synonym with less than or equal to half the number of word senses.

The Semantic Retrieval System uses different knowledge depending on the cell of the matrix that best describes the user, resulting in separate query processes and heuristics for each cell. As Table 2 shows, user profiles are only used when a query is delegated to another user/agent, or a user is a novice.

3.1 Type of User Profile

Researchers have been investigating how profiles could help queries for many years [9], but there remains no consensus on how to build, represent, or use a user profile [6], and the benefits have been inconsistent [10]. Table 3 summarizes the most well known approaches and the approach that we took with the Semantic Retrieval System.

Table 3. Summary of User Profile Techniques

Aspect of User Profile Research	Approaches in Literature	Example Reference	Approach in SRS
Types of user profiles	Knowledge-based vs. behavior based	[11]	Knowledge-based
	Personal vs. stereotype vs. community	[10]	Personal and stereotype
Approaches for creating user profiles	Direct/explicit (need user interaction) vs. Indirect/implicit (infer from user behavior)	[13], [4]	Direct and indirect
Approaches for representing user profiles	Keyword-based (e.g., vector) vs. Knowledge-based (e.g., rules, cases, frames, graph, tables)	[10]	Knowledge-based (frames)
Approaches for using user profiles	Statistical (e.g., cosine, Bayes) vs. Knowledge inference (e.g., rules, neural nets)	[8]	Knowledge-inference (rules)

User profiles are typically knowledge-based or behavior-based [11]. Knowledge-based profiles reflect user's knowledge in the form of semantics; behavior-based profiles store records of users actions, e.g., web sites visited. We use a knowledge-based approach because our objective is to reach a knowledgeable decision about the context of a user's query. Another distinction among profiles is whether the preferences are: 1) personal (i.e., individual); 2) stereotype (i.e., held by a class of individuals), or 3) community (i.e., held by an entire community) [10]. This research uses personal and stereotype approaches only because community preferences are less likely to be useful for a specific user and context. The Semantic Retrieval System chooses a suitable profile based on a user's level of domain knowledge. Although stereotypes can derive from any personal attribute, domain knowledge is a key one. Using domain knowledge requires the system to identify: (a) the domain(s) a user's query pertains to, and (b) the user's level of knowledge about the domain(s). As Table 2 shows, the system uses the following rules:

- If the user has a high level of domain expertise:
 - If the user writes her/his own query, do not use any profile,
 - If the user has an agent write the query, use the individual's personal profile,
- If the user has a low level of domain expertise:
 - If the user writes her/his own query, use the 'expert' stereotype profile,
 - If the user has an agent write the query, use the user's personal profile together with the 'expert' stereotype profile.

These rules first assume that users with more domain expertise have more stable preferences for what they want. Consequently, expert users are less likely to need to know what other users think about a topic. Second, users would rather know the preferences of experts than novices so a novice user uses the expert stereotype, rather than the novice stereotype. Our system uses these rules as defaults and users can change them.

3.2 User Profile Creation

The Semantic Retrieval System uses personal construct theory to guide profile creation. This theory assumes that people use a set of basic constructs to comprehend the world. An individual's construct system is personal because different people may use different constructs and may mean different things by the same constructs [5].

The system creates user profiles using both direct and indirect methods, as Table 3 shows. Because users are typically expert in some topics and novice in others, most users will use a combination of methods to build their profiles. The direct method is used for topics in which a person is expert. This is because a domain expert is more likely to have stable preferences about a domain and be more able to explicate her/his preferences. The first step is for the system to infer or for the user to instruct the system that he or she is an expert on a topic. The system will then require the user to manually create a profile on that topic by completing a form detailing the main constructs, key attributes of each construct, preferences about each attribute, and preferred instances. For example, a person might have a topic (e.g., Health), a construct (e.g., 'Doctor'), attributes (e.g., 'Location'), constraints (e.g., Location should be 'near to Atlanta'), and preferred instances (e.g., Dr Smith).

An indirect method is used to create novice profiles because the direct method is less likely to work if users' preferences are unstable and difficult to explicate. The first step in the indirect method is for the system to infer a user's topics and constructs by having the user complete practice queries for topics of interest. To expand the set of constructs and eliminate redundancy, the user's query terms are queried against WordNet and DAML ontology library to find synonyms and hypernyms. The system expands the constructs list by having the user rank the relevance of page snippets returned from queries. Highly ranked snippets are parsed for additional terms to add to the construct list. After finalizing the construct list, the system searches for relevant properties of each one by querying the ontology library. It also queries existing profiles from other users to find out if they use the same constructs and, if so, what properties they use. The system presents the user with a final list of properties from these sources (ontologies and existing user profiles) and asks the user to select relevant properties, add new ones if necessary, and rate their importance. Finally, the system requests that the user enter constraints for each property (e.g., Location 'near to Atlanta') as well as preferred instances, if any, of the construct (e.g., Dr Smith).

3.3 User Profile Representation

A frame representation is used to specify constructs (frames), attributes (slots), and constraints (slot values). There are two types of preferences as shown in Figure 3:

- *Global constraints* apply to all queries (e.g. pages returned must be in English).
- *Local constraints* apply based on the query context (location may be a relevant constraint for restaurants but not for online journals).

Frame: Global Constraint Slot Name: Language Slot Value: English Slot Name: Domain to search Slot Value: .com	Frame: Restaurant Slot Name: Food_type Slot Name: Cuisine Slot Name: Location Slot Name: Price	Value: Vegetarian Value: Italian Value: Atlanta Value: Inexpensive
---	--	---

Fig. 3. Global and Local Constraints

3.4 User Profile Use

We distinguish between *informational* and *action* queries. The query: *Mom needs to have a series of physical therapy sessions. Biweekly or something.... Set up the appointment* is an action query because it asks the ‘agent’ to ‘set up’ appointments. Our research is restricted to informational queries that simply retrieve information. Recall, from Table 2, that the Semantic Retrieval System follows one of four approaches depending on a user’s level of domain and lexical knowledge, and whether the user writes or delegates his/her query. The first step is to select the appropriate approach.

Step1: Identify approach:

- Heuristic 1: *Domain expertise*:

Assume that no query has yet been submitted. The user manually indicates his / her level of domain knowledge. Assume the user has high domain knowledge.

- Heuristic 2: *Query writer*:

No query has yet been submitted, so the user indicates whether s/he is the query originator. This is a delegated query for ‘Mom,’ so it resides in Cell 2, Table 2.

- Heuristic 3: *Language uncertainty*:

A user requires a lexicon if he or she submits a term that has a synonym with less than or equal to half the number of word senses. We first parse the query for stop words (via the list at http://dvl.dtic.mil/stop_list.pdf), action words (e.g., set), and identify phrases via WordNet. The query becomes:

series “physical therapy” sessions biweekly appointments.

Querying each term against WordNet, shows no synonyms or fewer word senses than its synonyms of the same sense. Thus, it can be inferred that the user has high language certainty.

The result of this step is that the system emphasizes user profiles over ontologies or lexicons to construct the query (see Table2, cell 2).

Step 2: Identify appropriate type of user profile:

The appropriate type of profile is a *personal* profile because the query has been delegated from an expert user.

Step 3: Identify terms from profile to increase query precision:

The system queries the user’s personal profile to identify relevant constraints. Assume ‘Mom’ had entered a profile with the preferences listed in Table 1. Figure 4 illustrates the resulting profile:

Frame: Global Constraint <i>Slot Name:</i> Language <i>Slot Value:</i> English <i>Slot Name:</i> Domain to search <i>Slot Value:</i> .com	Frame: Physical Therapist <i>Slot Name:</i> Treatment <i>Slot Name:</i> Provider <i>Slot Name:</i> Distance <i>Slot Name:</i> Service Level <i>Slot Name:</i> Appointment times	<i>Value:</i> Ultrasound <i>Value:</i> PSS Injury <i>Value:</i> Atlanta <i>Value:</i> High quality <i>Value:</i> Morning
---	--	--

Fig. 4. Profile for ‘Mom’ in Berners-Lee et al. Query

Given the query (*series “physical therapy” sessions biweekly appointments*), this step searches the profiles for any frame matching a query term. Only ‘Physical therapy’ matches a frame in the profile so the query becomes *series “physical therapy” ultrasound PSS injury Atlanta high-quality morning sessions biweekly appointments*.

Step 4: Execute query:

The system executes the expanded query. A manual test on Google returns zero results so the following heuristic is invoked.

- Heuristic 4: *Term reduction:*

If fewer than ten pages are returned and they do not include a relevant result, re-specify the query by removing constraints of lesser weight. Assume that users list attributes of constructs in descending order of importance. The query is iteratively re-run after deleting a term from those listed last in the profile. The query becomes:

“physical therapy” ultrasound PSS injury Atlanta

which returns two pages on Google:

- www.pssinjurycenter.com/PSSpatientForms.doc
- www.spawar.navy.mil/sti/publications/pubs/td/3138/td3138cond.pdf

After accounting for the global constraint, the second result is removed, leaving one page: www.pssinjurycenter.com/PSSpatientForms.doc. This is a relevant page for the query because it is the user’s preferred provider.

Step 5: Obtain feedback:

The user indicates whether the query was a success or a new query one is needed.

4 Prototype

A prototype of the profile module is being developed (Figure 5). Users specify queries in natural language. Java application code parses queries, inferences, adds context information, and constructs the search engine queries. The application interfaces with repositories that store User Profile and History information, lexicons (WordNet) and domain ontologies (DAML). The system interfaces with Google and Alltheweb.

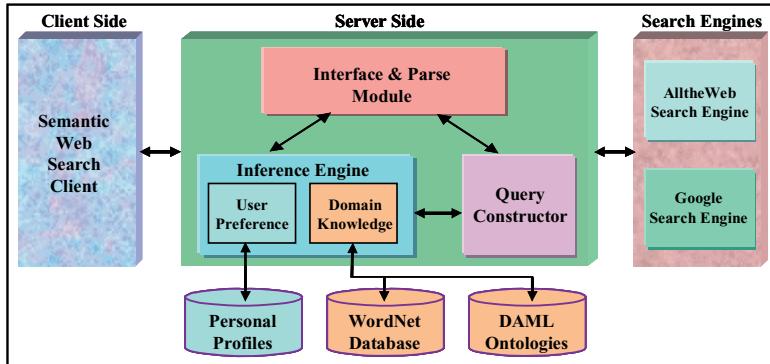


Fig. 5. Semantic Retrieval System Prototype Design

There are three modules. The *Interface & Parse Module* captures the user's query, parses it for nouns (noun phrases) and returns the part-of-speech for each term. From this, a baseline query is created. The *Local Knowledge Base & Inference Engine* interfaces with WordNet. Synonyms are first obtained for each noun in the initial query. Based on the user's selected synset, hypernyms and hyponyms for the selected sense of the term are obtained from the lexical database and the DAML library. The Local Knowledge Base & Inference Engine module is being implemented using Java Expert System Shell, Jess, (<http://herzberg.ca.sandia.gov/jess>). When processing a query, the local knowledge base is populated with domain knowledge and user preferences, expressed as Jess facts. For example, the domain knowledge base is instantiated with synonyms, hypernyms and hyponyms selected by the user. Jess rules reason about the context and retrieve relevant preferences using the base terms and synonyms, hypernyms and hyponyms. This contextual information is added to the knowledge base to identify the user's preferences; e.g., if the context is restaurants, then, the following rule will retrieve restaurant preferences.

```

Jess> (defrule get_restaurant_preferences "Get slots defined for food preferences)
  (phase: identifying preferences)
  ?cntxt <- (query-context restaurant)
  (restaurant_preferences (food_type ?x) (cuisine ?y) (domicile ?z) (driving_radius ?r)) =>
  (retract ?cntxt)
  (printout t "Restaurant Preferences" crlf)
  (printout t "FoodType: " ?x " Cuisine: " ?y " Domicile: " ?z " Driving Radius: " ?r crlf)
  (assert (Restaurant preferences retrieved)))

```

The *Query Constructor* expands the query, by adding synonyms, negative knowledge, hypernyms, and/or hyponyms, and personal preferences using several heuristics. The expanded query is submitted to the search engine and the results returned.

4.2 Sample Scenario

Assume a novice user enters the delegated query: "Find a restaurant for dinner." After parsing the user's input, the terms are displayed and the user can uncheck irrelevant

ones. The Inference Engine retrieves the word senses for each term from which the user identifies the appropriate sense. If none are selected, the module uses the query's base term; e.g., "dinner" has two senses. After selecting the appropriate word sense, the user initiates query refinement. For each term, the hypernyms and hyponyms corresponding to its word sense are retrieved from WordNet and the DAML ontologies. The user can identify hypernyms or hyponyms that best matches the query's intention to be added; e.g., the user might select "café." Additional contextual information is generated and used to search the user profile for preferences. For example, based on "café" and "meal", the system infers that the user is interested in having a meal in a restaurant and searches the user profile to retrieve preferences for restaurants and food habits. The restaurant preference frame shows that the user likes Indian Vegetarian food and does not like to drive more than 30 miles. The user lives in Atlanta. All preference information are added, but the user might not want to use some of it. (E.g., the user may relax the driving distance restriction.) The inference engine adds the additional information and the Query Constructor module generates the final query. Based on the word sense for each term, a synonym may be added with an OR condition. For example, "restaurant" is expanded with the first term in the user's selected synset (i.e., restaurant OR "eating house"). Negative knowledge is also added from the next highest synset that has a synonym to exclude unrelated hits. The user selected hypernyms and hyponyms are added as is the user preference information. For example, the Indian Vegetarian and Atlanta are added. The final refined query is then submitted to the search engine (Figure 6). The refined query for the restaurant example using Google's syntax is:

(restaurant OR "eating house") café "Indian vegetarian" Atlanta dinner meal

This query generates 79 hits. The user evaluates them by looking at the snippets or sites. Based on the snippets shown in Figure 6, the hits appear to be very relevant.

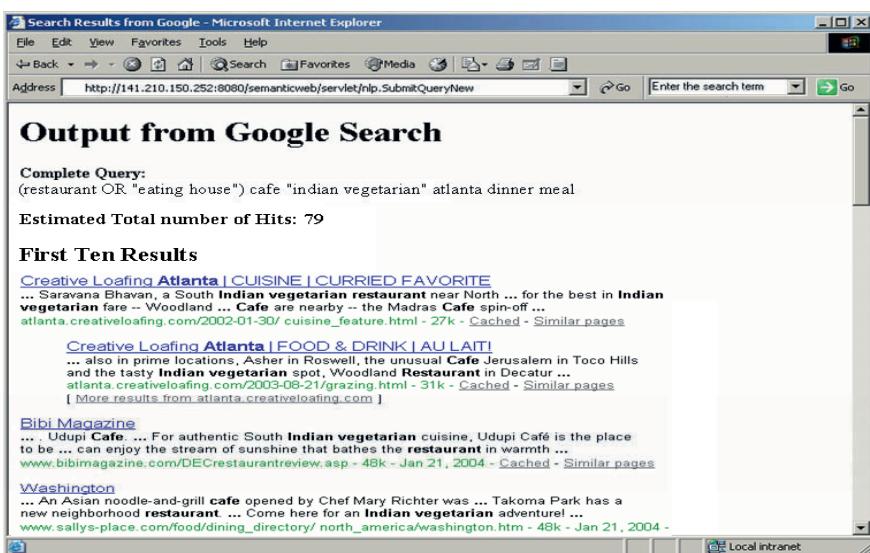


Fig. 6. Results from Google Search Engine

5 Testing

We manually simulated our approach to test whether it improves query relevance. We chose the context where user profiles should help most: when a domain novice delegates a query to another user (Table 2, cell 4). In this context, the system uses ontologies, lexicons, the novice's personal profile, and a stereotype profile. A restaurant domain was chosen and Figure 7 shows the user profiles and queries used for testing. One researcher created a personal profile and five queries while another created a stereotype profile for the restaurant domain. The stereotype profile was created based on the categories, attributes, and preferences listed in an online source of restaurant reviews (<http://www.accessatlanta.com>). Consistent with personal construct theory, the novice and expert frames differ in number of frames and types of slots.

Personal (Novice) Profile:		Stereotype (Expert) Profile:	
Frame: Restaurant		Frame: Restaurant	
Slots:	Values:	Slots:	Values:
- Maximum commute time - Food - Dining time - Usual number in party - Live music	- Less than 30 minutes - <i>Prefers</i> : American, French, Canadian, Italian; <i>Like</i> : Indian, Greek; <i>Dislike</i> : Seafood, fish, ethnic food - 7pm reservation - Four - No live music.	- Rating - Credit card - Parking	- Yes - Yes - Yes
Test Queries		Frame: Fine Dining Restaurant	
1. Find a restaurant that serves Canadian food. 2. Find a restaurant that serves PEI Mussels 3. Find a seafood restaurant that serves chicken 4. Find a restaurant that has a take-out area 5. Find a medium-expensive Italian restaurant in Dunwoody		Slots:	Values:
		- Menu - Service - Dress Code	- Excellent - Excellent - Formal
		Frame: Good Value Restaurant	
		Slots:	Values:
		- Menu - Service - Dress Code - Price	- Good - Good - Smart Casual - Reasonable

Fig. 7. Testing the query approach in the Restaurant domain

Table 4 details the results. Together, the knowledge sources provided many terms. Clearly, some terms such as '30 minutes' are less useful for a query. Our ongoing research is investigating whether inferences can be made about such statements to allow better query expansion. Nevertheless, by judiciously selecting terms from the list, we found that the relevance of each query could be improved, supporting the overall approach.

6 Conclusion

A methodology for incorporating user profiles into query processing has been proposed where user profiles are represented as frames. The profile information is applied to queries to improve the relevancy of the query results obtained. The methodology is being implemented in a prototype. Testing of the methodology shows that the current results are promising. Further research is needed to formalize the methodology, complete the prototype, ensure scalability, and further validation. The techniques developed for capturing, representing and using the user profile knowledge also need to be expanded to action queries.

Table 4. Summary of Testing Results

Q	Parsed Terms	Additional Query Terms Presented to User From...	Relevance of Top 10
1	Restaurant serves Canadian food	Lexicon: "serve up," provide Ontologies: café, "French Canadian" Profiles: less than 30 minutes, 7pm reservation, four people, -"live music", rating, credit card, parking	4 of 10 (Alltheweb); 6 of 10 (our system)
2	Restaurant serves PEI Mussels	Lexicon: "serve up," provide Ontologies: cafe Profiles: less than 30 minutes, 7pm reservation, four people, -"live music", rating, credit card, parking	2 of 10 (Alltheweb); 5 of 10 (our system)
3	Seafood restaurant serves chicken	Lexicon: "serve up," provide, poulet, volaille Ontologies: poultry, "Fish Food" Profiles: less than 30 minutes, (American or French or Canadian), -Seafood, -fish, -"ethnic food", 7pm reservation, four people, -"live music", rating, credit card, parking	4 of 10 (Alltheweb); 7 of 10 (our system)
4	Restaurant takeout area	Lexicon: None Ontologies: cafe Profiles: less than 30 minutes, (American or French or Canadian), -Seafood, -fish, -"ethnic food", 7pm reservation, four people, -"live music", rating, credit card, parking	2 of 10 (Alltheweb); 5 of 10 (our system)
5	Medium-expensive Italian restaurant Dunwood	Lexicon: average, moderate Ontologies: "European country" Profiles: less than 30 minutes, 7pm reservation, -Seafood, -fish, -"ethnic food", four people, -"live music", rating, credit card, parking, good menu, good service, smart casual	0 of 10 (Alltheweb); 6 of 10 (our system)

*The queries were run on the search engine <http://www.alltheweb.com>. Other search engines (e.g., Google) could not be used as they restrict the number of query terms (e.g., to ten terms).

References

1. Berners-Lee, T., J. Hendler, and O. Lassila, "The Semantic Web," in *Scientific American*, p. 1-19, 2001.
2. Borlund P. "The Concept of Relevance," *Journal of the American Society for Information Retrieval and Technology*, (54:10), 2003, pp. 91-103.
3. Burton-Jones, A., Storey, V.C., Sugumaran, V., and Purao, S. "A Heuristic-Based Methodology for Semantic Augmentation of User Queries on the Web," in *Proceedings of the 22nd International Conference on Conceptual Modeling*, 2003.
4. Claypool, M., Brown, D., Le, P., and Waseda, M. "Inferring User Interest," *IEEE Internet Computing*, Nov/Dec, pp. 32-39, 2001.
5. Cooper, A. "Individual Differences," 2nd Ed., Arnold Publishers, London, UK, 2002.
6. Godoy, D. and Amaldi, A. "A User Profiling Architecture for Textual-Based Agents," in *Revista Inteligencia Artificial de la Sociedad Espanola de Inteligencia Artificial (AEPIA)*, 2003.
7. Gruber, T.R. "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, 5(2):199-220, 1993.
8. Hanani, U., Shapira, B., and Shoval, P. "Information Filtering: Overviews of Issues, Research, and Systems," *User Modeling and User-Adapted Interaction*, 11(3):203-259, 2001.
9. Korfhage, R.R. "Query Enhancement by User Profiles," in *Proceedings of the Third Joint BCS and ACM Symposium*, pp. 111-122, 1984.

10. Kuflik, T., Shapira, B., and Shoval, B. "Stereotype-Based versus Personal-Based Filtering Rules in Information Filtering Systems," *Journal of the American Society for Information Science and Technology*, (54:3), pp. 243-250, 2003.
11. Middleton, S.E., Shadbolt, N.R., and De Roure, D.C. "Ontological User Profiling in Recommender Systems," *ACM Transactions on Information Systems*, Vol. 22, No. 1, pp. 54-88, 2004.
12. Spink, A., Wolfram, D., Jansen, B. J. and Saracevic, T. "Searching the Web: The Public and Their Queries," *Journal of the American Society for Information Science*, Vol. 52, No. 3, pp. 226-234, 2001.
13. Shepherd, M., Watters, C., Duffy, J., and Kaushik, R. "Browsing and Keyword-Based Profiles: A Cautionary Tale," *Proceedings of the 34th Hawaii International Conference on System Sciences*, IEEE Computer Society, January 3-6, 2001.

Deriving FrameNet Representations: Towards Meaning-Oriented Question Answering

Gerhard Fliedner^{1,2}

¹ DFKI GmbH, D-66123 Saarbrücken

² Computational Linguistics, Saarland University, D-66123 Saarbrücken

fliedner@coli.uni-sb.de

Abstract. We present a system for automatically annotating text with lexical semantic structures based on FrameNet, using a cascade of flat parsers with hand-crafted rules leading to a robust overall system.

This translation process is eventually to be employed in a Question Answering system to allow a more meaning-oriented search for answers: We want to explore the possibilities of a match of the FrameNet representations of the document base with that of the user's query. This will allow a more directed matching than the often-used approach of first retrieving candidate documents or passages with techniques based on surface words and will bring together ideas from Information Extraction, Question Answering and Lexical Semantics. FrameNet allows us to combine words in the text by 'semantic valency', based on the idea of thematic roles as a central aspect of meaning. This description level supports generalisations, for example over different words of one word field.

First results of the FrameNet translation process are promising. A large scale evaluation is in preparation. After that, we will concentrate on the implementation of the Question Answering system. A number of implementation issues are discussed here.

1 Overview

We present a system for automatically annotating German texts with lexical semantic structures, namely FrameNet, using a cascade of flat parsers. This process is eventually to form the core of a meaning-oriented Question Answering (QA) system where both the text corpus to be searched and the user's questions to the system are automatically translated into FrameNet representations and matching is done directly on these structures. This is ongoing work within the Collate project: While the greater part of the FrameNet translation process has been implemented and is described here in more detail, the QA system is still in its design phase. We will give a sketch of some design issues.

Most QA systems today use a different approach [1]: After processing the user's question, a document (or sometimes passage) search is done using keywords from the question, often complemented by semantically related words (query expansion). The actual search is thus done based on surface words, mostly using indexing techniques. The retrieved candidate documents or passages are

then processed either statistically [2] or with ‘deeper’ linguistic methods [3,4,5]. The answer is generated from the passage that was rated most relevant, either as a text snippet or (if a linguistic analysis was performed) using generation techniques on the analysis results.

Using surface word-based Information Retrieval (IR) techniques to find candidate documents has become state-of-the-art after several attempts more or less failed to introduce ‘deep’ linguistic processing into QA in the 1970’s. The IR approach generally leads to very efficient processing. However, these systems have a number of potential disadvantages, such as imperfect precision and high reliance on answer redundancy [6].

In contrast, matching structured representations has in the past mainly been employed in natural language database front-ends (for an overview see [7]). These front-ends allow a user to query structured databases in natural language, ideally without any knowledge about the design or internal working of the database. Since the front-ends are mostly tailored to the respective task, they can reach a high overall performance. On the other hand, they are more or less domain dependent and do not scale up well in general. In addition, they are fitted to one specific database and cannot simply accommodate new, raw data (especially no unstructured text).

Another related field of work is Information Extraction (IE). Here, text collections are processed to fill pre-defined templates with specific information. This normally takes the form of identifying interesting entities (persons, organisations etc.), identifying relations between them (e.g., EMPLOYEE_OF), and finally filling general scenario templates (e.g., describing a launch of a satellite in detail). Well-known tasks in IE have been defined for the Message Understanding Conferences [8]. Both rule-based and machine learning approaches have been applied to these tasks in the past. The templates defined for IE tasks tend to be specific and detailed, making it necessary to customise at least the template filling module for each domain. FASTUS is a prominent example of such a system [9].

We want to make use of the idea of pre-processing the text to enrich it with more structure. By doing the annotation process ‘off-line’, i.e., at document indexing time, the actual search can efficiently be done at retrieval time over structured data. As basis for the annotation format, we have chosen the lexical semantic structures of FrameNet, using its concept of semantic valency to abstract away over linguistic issues on the text surface, such as wording or systematic differences such as active/passive. The idea is to use the translation process and the resulting FrameNet structures as a representation of both the texts that are searched and the user’s question to the system and to match these representations as part of a more meaning oriented search mechanism: If we know the exact relation between two entities in the text and can directly search that relation, then we can use this knowledge to answer questions about them in a more principled way than if we only observe that they co-occur in a text (as with bag-of-words techniques).

Our goal thus lies between the pure bag-of-words-based search and the systems for IE and natural language database front-ends described above in that it

will be as domain-independent as possible on the one hand and will, on the other hand, allow finding answers to questions in a more meaning-oriented way. Such a result will bring together IE and QA. This is achieved by using FrameNet structures that describe general natural language structures and not templates that are defined by an underlying database or by a specific IE task: Sometimes, even human annotators seem to have had difficulties in filling IE scenario templates from given texts [8].

We believe that our approach, that is linguistically grounded and thus more focused on the language of the text collection than on abstract information structures, can be more reliably derived automatically and still offers a reasonable structured representation to be searched. Our approach has the additional advantage that an answer generation becomes comparatively easy: Through the retrieval process, we have direct access to a structured representation of the original text. This can, together with the answer type, be used to generate a natural language answer.

This paper is organised as follows: First we give a short introduction to FrameNet and argue for the usefulness of a FrameNet annotation for linguistic tasks. We describe our system set-up for the derivation of FrameNet structures and give an example of the processing. Then we sketch some challenges that still must be solved in integrating this parser into a full-fledged QA system. Finally, we conclude by summarising the current state of the project and giving an outlook.

2 FrameNet for Semantic Annotation

FrameNet is a lexical database resource containing valency information and abstract predicates. English FrameNet is developed at the ICSI, Berkeley, CA¹ [10,11]. Development of a German FrameNet is currently underway within the context of the SALSA project² [12]. Here, a large corpus of German newspaper texts is annotated with FrameNet structures in a boot-strapping fashion, beginning with manual annotation of sub-corpora, while techniques are developed for (semi-) automatic annotation that will eventually speed up annotation substantially. Thus, completing the FrameNet database with valency information and annotating the corpus proceed in an interleaved fashion.

Intuitively, a frame in FrameNet is a schematic representation of a situation involving participants, props etc. Words are grouped together in frames according to semantic similarity, a number of frames forming a domain. For each frame, a small number of frame elements is defined, characterising semantic arguments belonging to this frame. For example, the English verbs *buy* and *sell* belong to frames in the COMMERCE domain. Among others, a frame element BUYER is defined for them. When annotating documents using FrameNet, a BUYER would always be labeled as such, no matter that its syntactic relation to the verb *buy*

¹ framenet.icsi.berkeley.edu/~framenet/

² *The Saarbrücken Lexical Semantics Annotation and Analysis Project*, www.coli.uni-sb.de/lexicon/

or *sell* may be different (e.g., subject *vs.* indirect object). A FrameNet representation of ‘Peter sells the book to Mary for £2.’ is shown in Ex. 1.

SELLER:	<i>Peter</i>
GOODS:	<i>book</i>
BUYER:	<i>Mary</i>
MONEY:	£2

Example 1.

FrameNet is used here to describe semantic roles and thus the relations between participants and objects in a situation as predicate-argument-structures. Words that are semantically similar will receive comparable descriptions with the same role labels. This not only holds for synonyms, but also for antonyms and converse relations (such as *buy* and *sell*), and across parts of speech.

This way of semantic grouping differs from ontologies like WordNet and its German version, GermaNet [13], that mainly concentrate on notions of hyponymy and hypernymy. They do not contain information on the semantic valency of words, i.e., the relations described above. However, semantic relations are central to capturing the meaning of texts, as argued above.

FrameNet also differs from other representations of semantic valency like tectogrammatical structures [15] or PropBank [16]: In these approaches, the buyer in an act of selling would, e.g., be described either as the Addressee or the Actor by tectogrammatical structures, or as an Arg0 or Arg2 by PropBank, depending on whether the words *buy* or *sell* are used (Addressee/Arg2 of the selling, but Actor/Arg0 of the buying). In FrameNet, the roles are characterised with respect to the frame, therefore the buyer is a BUYER, no matter which words are used.

This representation is therefore especially suited for applications where the surface wording is less important than the contents. This is the case for Information Management systems: In IE and IR, especially in QA, it is far more important to extract or find the right contents; differences in wording are more often than not a hindrance in this process.

For a QA system, we plan to use the FrameNet translation to annotate the text basis to be used for QA ‘off-line’. The FrameNet structures are then stored in a way that allows efficient access. In the actual query process, the user’s questions are again translated by the same FrameNet translation process, supported by a question type recognition module.

Matching against FrameNet structures instead of just words in an index will, e.g., allow to find an answer to the question ‘Who bought Mannesmann?’, no matter if the relevant text passage originally reads ‘Vodafone took over Mannesmann in 2000.’, ‘Mannesmann was sold to Vodafone.’, or ‘Vodafone’s purchase of Mannesmann...’ This will not only increase recall (because the wording must not be matched exactly), but also precision (because texts in which the words *bought* and *Mannesmann* co-occur by chance do not match).

As we have described above, the off-line indexing process described here is in principle an IE task. However, as it is based on the semantic valency of words

here and not on abstract IE templates and thus closer to the text itself, it is less specialised and domain-dependent and expected to scale up better. We have started our work on a limited domain (business news texts). Exemplary tests with other texts (law, politics) have shown, however, that the approach is not limited by the domain.

3 Deriving FrameNet Structures from Text

In the previous section, we have argued for the usefulness of a text representation based on semantic valency. We have claimed that such a representation may be automatically derived from texts, as it is closely enough related to the text. We have implemented a system for deriving a FrameNet structure from German texts, based on flat parsing modules. Section 4 gives an example of the processing.

Our system for annotating German texts with FrameNet structures uses a cascaded parsing process. Different parsers build on the results of each other to finally assign FrameNet structures to the text. These parsers focus only on recognising certain linguistically motivated structures in their respective input and do not try to achieve spanning parses. All parsers may leave ambiguities unresolved in the intermediate results, so that a later processing step may resolve them. This general technique was introduced under the name of easy-first-parsing in [17]. It generally leads to a more robust overall system.

The input text is first tokenised and analysed morphologically. We employ the Gertwol two-level morphology system that is available from Lingsoft Oy, Helsinki. Gertwol covers the German morphology with inflection, derivation and compounding and has a large lexicon of approximately 350,000 word stems.

Next is a topological parser. German sentences have a relatively rigid structure of topological fields (*Vorfeld*, left sentence bracket, *Mittelfeld*, right sentence bracket, *Nachfeld*) that helps to determine the sentence structure. By determining the topological structure many potential problems and ambiguities can be resolved: The overall structure of complex sentences with subclauses can be recognised and the main verbs and verb clusters formed by modal verbs and auxiliaries can be identified with high precision. The topological parser uses a set of about 300 hand-crafted context-free rules for parsing. Evaluation has shown that this approach can be applied successfully to different sorts of texts with both recall and precision averaging 87 % (perfect match) [18].

Named Entity Recognition (NER) is the next step in the processing chain. We use a method based on hand-crafted regular expressions. At the moment, we recognise company names and currency expressions, as well as some person names. Recognition of company names is supported by a gazetteer with several thousand company names in different versions (such as *ABB*, *ABB Asea Brown Boveri* and *ABB Group*). The NE grammars and the gazetteer have been developed as part of a multi-lingual IE system within our project [19]. In an evaluation, the module has reached state-of-the-art results (precision of up to 96 %, recall of up to 82 % for different text sorts).

In the long run, we consider enhancing the NE recogniser by methods for the automatic recognition of new NE patterns. As there are no out-of-the-box NERs for German that we are aware of, and as systems based entirely on machine learning techniques as the ones used in the multilingual named entity recognition shared task at CoNLL 2003 [20] need a relatively large amount of annotated training data, we think that a bootstrapping approach will prove the best option. Several methods have been proposed for this. Among them, especially the ‘learn-filter-apply-forget’ method [21] has been successfully applied for German.

Named entity recognition is followed by a chunker for noun phrases (NPs) and prepositional phrases (PPs). This chunker was originally built for grammar checking in German texts [22]. It uses extended finite state automata (FSA) for describing German NPs/PPs. Though simple German NPs have a structure that can be described with FSA, complex NPs/PPs show self-embedding of (in principle) arbitrary depth: Attributive adjectives, for example, can take arguments in German, leading to centre self-embedding. The extension has been added exactly to accommodate these structures. This allows our system to analyze NPs like *[das [vom Konkurrenten]PP übernommene Unternehmen]*_{NP} (‘the by the competitor taken-over company’, the company taken over by the competitor).

One important feature of our system is that the results of the NE recogniser directly feed into the NP chunker. This allows us to handle an NE as an NP or N', so that it can form a part of more complex NP. Among other phenomena, modification with adjectives and coordination can thus be handled. One example of such a complex NP: *ein Konsortium aus [Siemens AG]*_{NE} *und mehreren Subunternehmern* (a consortium of Siemens Ltd. and a number of sub-contractors).

We have evaluated our original NP chunker without NE recogniser by using 1,000 sentences from the annotated and manually controlled NEGRA corpus [23] as a gold standard to evaluate the bracketing of NPs. Of the 3,600 NPs in this test corpus, our system identified 92 %. Most of the remaining 8 % were due to unrecognised named entities. Of the recognised NPs, the bracketing was identical with the NEGRA corpus in 71 % of the cases. This relatively low figure is caused mostly by the fact that post-nominal modifiers, in contrast to the NEGRA corpus, are not attached to the NP by our system (as this is done by later processing steps), accounting for about one third of the errors.

The results of the previous stages are put together into one overall structure. We have called this structure PReDS (Partially Resolved Dependency Structure, [18]). PReDS is a syntacto-semantic dependency structure that retains a number of syntactic features (like prepositions of PPs) while abstracting away over others (like active/passive). It is roughly comparable with Logical Form [5] and Tectogrammatical Structures [15]. PReDS is derived from the structures built so far using context-free rules.

In a last step, the resulting PReDS structure is translated into FrameNet structures [24]. This translation uses weighted rules matching sub-trees in the PReDS. The rules can be automatically derived from a preliminary version of a FrameNet database containing valency information on an abstract level (e.g. using notions like deep subject to avoid different descriptions for active and

passive on the one hand, but keeping prepositions as heads of PPs on the other hand). The FrameNet coverage for German is yet exemplary, but will grow with the increasing coverage of the German FrameNet lexicon.

One additional processing component has not yet been integrated into the system, namely an ontology providing a hierarchy of semantic sortal information that can be used for describing selectional preferences on frame elements. So far, we have found that the so-called ‘core frame elements’, that must necessarily be present, are in most cases well-identifiable in syntax as, for example, subjects or objects, whereas ‘non-core elements’ such as TIME or LOCATION can often not be recognised by syntactic features alone. For example, both TIME and LOCATION can be given in a PP_{in}, as ‘in the last year’ (TIME) or ‘in the capital’ (LOCATION). Using sortal information (we plan to employ GermaNet) will help to resolve cases that so far must remain underspecified.

So far, only walk-through evaluations for single sentences have been conducted for our system. In these, currently about three quarters of the sentences of a test corpus of business news receive a PReDS representation as a basis for the FrameNet annotation. For FrameNet frames, the core frame elements are mostly correctly assigned, whereas non-core elements are less successfully handled.

A quantitative evaluation is planned to start shortly. We plan to focus on two ‘gold standard’ evaluations using annotated corpora. For the evaluation of the PReDS structures we are currently looking into the possibility of using as a gold standard the German TIGER newspaper corpus of 35,000 sentences that has a multi-level annotation including grammatical functions [25]. The evaluation would be based on the grammatical functions. A suitable scheme for dependency-oriented parser evaluation has been described in [26].

To evaluate the FrameNet annotation (i. e., end-to-end evaluation), we plan to use the finished parts of the SALSA corpus of FrameNet for German [12] in a similar way. Due to the still small coverage of the SALSA corpus, this may have to be complemented by manual annotation and evaluation of a certain amount of text. Based on the first walk-through evaluations, we would expect our system to reach precision and recall values around the 60 % mark. This would be on par with the results of related work, shortly described in the following.

For English FrameNet, a system for automatically annotating texts with FrameNet structures has been described [27]. It uses machine learning techniques and thus needs to be trained with a corpus that is annotated with FrameNet structures. The authors report a recall of 61 % and a precision of 65 % for the text-to-frame element labelling task. As this approach needs a large amount of training data (the authors have used the FrameNet corpus with 50,000 labeled sentences), we could not easily transfer it to German, where the SALSA FrameNet corpus is only now under development.

An approach partly similar to ours has been described for annotating Czech texts in the Prague Dependency Tree Bank with tectogrammatical (i. e., semantic) roles [28]. The author uses a combination of hand-crafted rules and machine learning to translate a manually annotated syntactic dependency structure into the tectogrammatical structures. He reports recall and precision values of up to

100 % and 78 %, respectively, or, with different system settings of 63 % and 93 %. However, this approach does not start from text, and therefore only shows how well the translation from dependency structures to semantic roles can do.

4 One Example

As an example we have chosen a sentence originally taken from our corpus of German business news texts (Ex. 2). The parse results for the different layers are shown in Fig. 1.

Lockheed hat von Großbritannien den Auftrag für 25
 Lockheed has from Great Britain the order for 25
 Transportflugzeuge erhalten.

Example 2. transport planes received.

'Lockheed has received an order for 25 transport planes from Great Britain.'

FrameNet										
PReDS										
Topologic & Chunker										
Text										
Morph										

Fig. 1. Example sentence with parse results (simplified from *Süddeutsche Zeitung*, 2 January 95)

In constructing the PReDS, the different tools contribute in the following way: The topological parser is used to recognise the split verb *hat...erhalten*. The NE recogniser finds *Lockheed*, whereas the other NPs are detected by the NP chunker. Note that in the construction of the PReDS all uncertain decisions are postponed for later processing steps. For example, the PPs always receive low attachment by default: The PP *für* is thus attached to the preceding NP, though syntactically a PP *für* might also be a verb modifier, e.g., expressing a beneficiary (like a free dative).

A number of FrameNet structures are constructed based on the PReDS. Two of them are shown in Exs. 3 and 4. These structures are derived from the PReDS by lexical rules containing different possible syntactic patterns for the targets.

Example 3.

GETTING
TARGET: <i>erhalten</i>
DONOR: <i>Großbritannien</i>
RECIPIENT: <i>Lockheed</i>
THEME: <i>Auftrag</i>

Example 4.

REQUEST
TARGET: <i>Auftrag</i>
MESSAGE: <i>25 Transportflugzeuge</i>

The underlying rule deriving the GETTING frame is shown in Ex. 5. The valency information expressed in the rules allows the derivation process to repair incorrect default choices made during the PReDS construction. For example, a PP that received the standard low attachment might be changed from an NP modifier into a verb complement, given the correct verb valency.

Example 5.

GETTING
TARGET: <i>erhalten-verb</i>
RECIPIENT: Deep-Subject
THEME: Deep-Object
[DONOR: PP-von]
...

The process described here can only derive FrameNet relations that are expressed syntactically in some way (e.g., verb-object-relation). The next step to be implemented is the merging of information from different basic frames. In this example one would like to infer that the order is given to Lockheed by Great Britain by transferring the information from the GETTING to the REQUEST.

Our experience so far has shown that there are relatively many cases like the one shown in the example where not all frame elements are part of the target word's syntactic valency domain. In these cases, information must be transferred from other frames. This can partly be compared with template fusion in information extraction. Some of the cases are very systematic. We will investigate both hand-coding and learning these rules from the annotated corpora.

5 Implementation Issues of the QA System

In the previous sections, we have described how FrameNet representations are derived from texts. We now turn to the question how these structures may actually be used in question answering. These issues are taken up in [29].

As both the document collection and the user's question are translated into FrameNet structures, the actual matching should in principle be straightforward: In question processing, the 'focus' of the question (i.e., the word or concept that the user asked for) would be replaced by a wild-card. Then the search would be a lookup in the list of frames generated by the text annotation. Processing

the question to find the focus and expected answer type has proven to be an important issue in QA systems [4,30]. We plan to use patterns over FrameNet structures to match question types similar to the approaches described there.

For the actual search, we need to store the resulting FrameNet structures in a way to allow efficient access, preferably in a relational database. This should ideally support searching for hyponyms and hypernyms, allowing, e.g., for a question that contains a *buy* not only to match the COMMERCE_BUY frame, but also the more general GETTING frame that contains words like *obtain* or *acquire*, and vice versa. This information is present both in FrameNet and GermaNet. Methods for mapping such matches onto database queries have been implemented for XML structures [31]. This must be complemented by introducing underspecified ‘pseudo-frames’ for words and concepts that are not yet covered by FrameNet. In cases where a question is translated into more than one relation (e.g., ‘What is the size of the order given to Lockheed by Great Britain?’, where first the order must be identified and then its size must be found), this should in general translate into a database join. However, there will probably be cases where a directed inference step is required to bridge the gap between question and text representation.

Introducing a directed inferencing step in this process would help to find answers that would otherwise be lost (especially in cases of a mismatch in the granularity between question and text representation, as mentioned above). Examples of directed inferences are discussed in [3]. We are currently investigating the possibility of automatic inferencing over the FrameNet structures similar to the approach taken there.

6 Conclusions and Outlook

We have presented a system for automatically adding FrameNet structures to German texts using a cascade of flat parsers. Distributing the task of parsing over different parsers each handling only one single linguistic layer helps to make our approach robust and scalable. We have argued that FrameNet annotation adds a level of information that is especially useful for tasks like IE/IR, as semantically related words belonging to the same frame receive a comparable representation. We plan to use our system in a QA system using direct matching of FrameNet representations of both the user’s question and the texts to be searched. After the system evaluation described in Sec. 3, we plan to implement the QA system described in Sec. 2. Several open questions have been presented in Sec. 5.

Other envisaged uses include annotating Intranet/Internet pages with FrameNet annotation and then translating the resulting structures into one of the formalisms associated with the Semantic Web. It has been shown for English FrameNet that this translation is relatively straightforward [32]. This is, however, constrained by annotation speed: At the moment, it takes a few seconds to annotate complex sentences. This will be sufficient to annotate collections of tens of thousands of documents (provided that the rate of document change is

comparatively low). It will, however, not yet be suitable for a reasonably up-to-date Internet service with millions of pages.

An important part of the development of the overall QA system described in Sec. 2 will be an evaluation to see if a FrameNet based system can really help to improve the performance of a QA system compared to one using bag-of-words techniques measurably. This is described in more detail in [33].

Acknowledgements. We would like to thank the anonymous reviewers for NLDB for their valuable comments on a draft version of this paper.

The work described in this paper is ongoing work within the Collate project (*Computational Linguistics and Language Technology for Real Life Applications*), conducted by DFKI GmbH (partly jointly with Saarland University), funded by the German Ministry for Education and Research, Grant numbers 01 IN A01 B and 01 IN C02.

References

1. Hirschman, L., Gaizauskas, R.: Natural language question answering: the view from here. *Natural Language Engineering* **7** (2001) 275–300
2. Zhang, D., Lee, W.: A language modeling approach to passage question answering. In: Proceedings of TREC 2003, Gaithersburg, MD (2003)
3. Moldovan, D., Harabagiu, S., Girju, R., Morarescu, P., Lacatusu, F., Novischi, A., Badulescu, A., Bolohan, O.: LCC tools for question answering. In: Proceedings of TREC 2002, Gaithersburg, MD (2002)
4. Hermjakob, U.: Parsing and question classification for question answering. In: ACL 2001 Workshop “Open-Domain QA”, Toulouse, France (2001)
5. Elworthy, D.: Question answering using a large NLP system. In: Proceedings of TREC 9, Gaithersburg, MD (2000)
6. Light, M., Mann, G.S., Riloff, E., Breck, E.: Analyses for elucidating current question answering technology. *Natural Language Engineering* **7** (2001) 325–342
7. Copestake, A., Sparck Jones, K.: Natural language interfaces to databases. *Knowledge Engineering Review* **4** (1990) 225–249
8. Marsh, E., Perzanowski, D.: MUC-7 evaluation of IE technology: Overview of results. In: Proceedings of MUC-7, Washington, DC (1998)
9. Appelt, D.E., Hobbs, J.R., Bear, J., Israel, D., Kameyama, M., Tyson, M.: SRI International FASTUS system MUC-6 results and analysis. In: Proceedings of MUC-6, Columbia, MD (1995) 237–248
10. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of COLING 98, Montreal, Canada (1998)
11. Johnson, C.R., Fillmore, C.J., Petrucc, M.R.L., Baker, C.F., Ellsworth, M., Ruppenhofer, J., Wood, E.J.: FrameNet: Theory and practice. Internet: <http://www.icsi.berkeley.edu/~{}framenet/book/book.html> (2002)
12. Erk, K., Kowalski, A., Pinkal, M.: A corpus resource for lexical semantics. In: Proceedings of IWCS 2003, Tilburg (2003)
13. Kunze, C., Lemnitzer, L.: Germanet – representation, visualization, application. In: Proceedings of LREC 2002, Las Palmas (2002) 1485–1491
14. Abeillé, A., ed.: Building and Using Parsed Corpora. Kluwer Academic Publisher, Dordrecht (2003)

15. Böhmová, A., Hajič, J., Hajičová, E., Hladká, B.: The Prague Dependency Treebank. A three-level annotation scenario. [14] chapter 7 103–127
16. Palmer, M., Gildea, D., Kingsbury, P.: The Proposition Bank: An annotated corpus of semantic roles. Submitted to: Computational Linguistics (2004)
17. Abney, S.: Partial parsing via finite-state cascades. In: Proceedings of ESSLLI Workshop on Robust Parsing, Prague (1996) 8–15
18. Braun, C.: Parsing German text for syntacto-semantic structures. In: Workshop “Prospects and Advances in the Syntax/Semantics Interface”, Nancy, France (2003) 99–102
19. Bering, C., Drożdżyński, W., Erbach, G., Guasch, C., Homola, P., Lehmann, S., Li, H., Krieger, H.U., Piskorski, J., Schäfer, U., Shimada, A., Siegel, M., Xu, F., Ziegler-Eisele, D.: Corpora and evaluation tools for multilingual named entity grammar development. In: Proceedings of Corpus Linguistics Workshop “Multilingual Corpora”, Lancaster (2003) 42–52
20. 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 (2003) 142–147
21. Volk, M., Clematide, S.: Learn-filter-apply-forget. Mixed approaches to named entity recognition. In: Proceedings of NLIS 2001, Madrid (2001)
22. Fliedner, G.: A system for checking NP agreement in German texts. In: Proceedings of ACL Student Workshop, Philadelphia, PA (2002)
23. Brants, T., Skut, W., Uszkoreit, H.: Syntactic annotation of a German newspaper corpus. In: Proceedings of ATALA Treebank Workshop, Paris (1999) 69–76
24. Fliedner, G.: Tools for building a lexical semantic annotation. In: Workshop “Prospects and Advances in the Syntax/Semantics Interface”, Nancy, France (2003) 5–9
25. Brants, S., Dipper, S., Hansen, S., Lezius, W., Smith, G.: The TIGER Treebank. In: Proceedings of the Workshop on Treebanks and Linguistic Theories, Sozopol, Bulgaria (2002)
26. Carroll, J., Minnen, G., Briscoe, T.: Parser evaluation. Using a grammatical relation annotation scheme. [14] chapter 17 299–316
27. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. Computational Linguistics **28** (2002) 245–288
28. Žabokrtský, Z.: Automatic functor assignment in the Prague Dependency Treebank. A step towards capturing natural language semantics. Master’s thesis, Department of Computer Science, Czech Technical University, Prague (2001)
29. Fliedner, G.: Towards using FrameNet for question answering. To Appear in: Proceedings of LREC Workshop “Building Lexical Resources” (2004)
30. Harabagiu, S., Moldovan, D., Pașca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Gîrju, R., Rus, V., Morărescu, P.: Falcon: Boosting knowledge for answer engines. In: Proceedings of TREC 9, Gaithersburg, MD (2002)
31. Schenkel, R., Theobald, A., Weikum, G.: Ontology-enabled XML search. In Blanken, H., Grabs, T., Schek, H.J., Schenkel, R., Weikum, G., eds.: Intelligent Search on XML Data. Applications, Languages, Models, Implementations, and Benchmarks. Number 2818 in LNCS. Springer, Berlin (2003) 119–131
32. Narayanan, S., Fillmore, C.J., Baker, C.F., Petrucci, M.R.L.: FrameNet meets the semantic web: A DAML+OIL frame representation. In: Proceedings of the National Conference on AI, Edmonton, Canada (2002)
33. Fliedner, G.: Issues in evaluating a question answering system. To Appear in: Proceedings of LREC Workshop “User-Oriented Evaluation” (2004)

Lightweight Natural Language Database Interfaces

In-Su Kang¹, Seung-Hoon Na¹, Jong-Hyeok Lee¹, and Gijoo Yang²

¹ Division of Electrical and Computer Engineering,
Pohang University of Science and Technology (POSTECH),
Advanced Information Technology Research Center (AITrc),
San 31, Hyoja-dong, Nam-gu, Pohang, 790-784, Republic of Korea
{dbaisk, nsh1979, jhlee}@postech.ac.kr

² Department of Information and Communication Engineering,
Dongguk University, Pil-dong, 3 Ga 26, Seoul, 100-715, Republic of Korea
gjyang@dgu.ac.kr

Abstract. Most natural language database interfaces suffer from the translation knowledge portability problem, and are vulnerable to ill-formed questions because of their deep analysis. To alleviate those problems, this paper proposes a lightweight approach to natural language interfaces, where translation knowledge is semi-automatically acquired and user questions are only syntactically analyzed. For the acquisition of translation knowledge, first, a target database is reverse-engineered into a physical database schema on which domain experts annotate linguistic descriptions to produce a *pER* (physically-derived Entity-Relationship) *schema*. Next, from the *pER schema*, initial translation knowledge is automatically extracted. Then, it is extended with synonyms from lexical databases. In the stage of question answering, this semi-automatically constructed translation knowledge is then used to resolve translation ambiguities.

1 Introduction

Natural language database interfaces (NLDBI) allow users to access database data in natural languages [2]. In a typical NLDBI system, a natural language question is analyzed into an internal representation using linguistic knowledge. This internal representation is then translated into a formal database query by applying translation knowledge that associates linguistic constructions with target database structures. Finally, the database query is delivered to the underlying DBMS.

Translation knowledge, which is created for a new target database, is necessarily combined with linguistic knowledge. Previous works are classified according to the extent that these two types of knowledge are connected: monolithic, tightly coupled, and loosely coupled approaches. In monolithic approaches, anticipated question patterns are associated with database query patterns. Question patterns are defined at the lexical level, so they may over-generalize the question meaning. Tightly coupled approaches hard-wire translation knowledge into linguistic knowledge in the form of semantic grammars [6,13,12]. These two approaches require modification of the two

kinds of knowledge in porting to a new domain. So, NLDBI researchers have concentrated on minimizing the connection of the two types of knowledge in order to improve transportability.

Loosely coupled approaches entail further division. Syntax-oriented systems [3,4,8] analyze questions up to a syntactic level. Logical form systems [14,5,1] interpret a user question into a domain-independent literal meaning level. In these approaches, translation knowledge is applied after analysis. Thus, transporting to new database domains does not need to change linguistic knowledge at all, only tailor translation knowledge to new domains. Even in this case, however, it is nontrivial to describe translation knowledge. For example, syntax-oriented systems have to devise conversion rules that transform parse trees into database query expressions [2], and logical form systems should define database relations for logical predicates. In addition, creating such translation knowledge demands considerable human expertise in AI/NLP/DBMS and domain specialties.

Moreover, in state-of-the-art logical form approaches, a question should successfully undergo tokenization, tagging, parsing, and semantic analysis. However, these entire analysis processes are vulnerable to ill-formed sentences that are likely to occur in an interactive environment, where users are prone to type their request tersely.

In order to automate translation knowledge acquisition, and to make a robust NLDBI system, this paper proposes a lightweight NLDBI approach, which is a more portable approach because its translation knowledge does not incorporate any linguistic knowledge except words. Note that, in loosely coupled approaches, translation knowledge encodes some linguistic knowledge, such as parse trees, or logical predicates. Our lightweight approach features semi-automatic acquisition and scalable expansion of translation knowledge, use of only syntactic analyzers, and incorporation of information retrieval (IR) techniques for conversion of question nouns to domain objects.

The remainder of this paper is as follows. The next section defines the terminologies used in this paper. Section 3 describes translation difficulties in NLDBI, and Section 4 through 6 explains a lightweight NLDBI approach using examples. Discussions and concluding remarks are given in section 7.

2 Terminologies

In this paper, a **domain class** refers to a table or a column in a database. A **domain class instance** indicates an individual column value. For example, suppose that a database table T_customer has a column C_name. Then, T_customer and C_name are called domain classes. If C_name has ‘*Abraham Lincoln*’ as its value, this value is called a domain class instance. A **class term** is defined as a lexical term that refers to a domain class. A **value term** means a term that indicates a domain class instance. For instance, the word ‘*customer*’ in a user question is a class term corresponding to the above domain class T_customer. The word ‘*Lincoln*’ is a value term referring to the above domain class instance ‘*Abraham Lincoln*’.

3 Translation Difficulties in NLDBI

When translating a question noun into its correct domain class, main difficulties are due to paraphrases and translation ambiguities, which correspond to M-to-1 and 1-to-N mappings between linguistic expressions and domain classes (or domain class instances), respectively. For example, several different paraphrases (e.g., *professor*, *instructor*, *a person who teaches*, etc.) may refer to the same domain class, and the same linguistic expression (e.g., *year*) may refer to many different domain classes, such as entrance year, grade year, etc.

In NLDBI, paraphrases can occur as class terms or value terms. Class term paraphrases correspond to synonyms or hypernyms. Value term paraphrases tend to occur in abbreviations, acronyms, or even substrings. For example, *MS*, *JFK*, and *Boeing* can refer to '*Microsoft*', '*John F. Kennedy airport*', and '*Boeing xxx-xxx*', respectively. So we should support partial matching between question value terms and domain class instances. In other words, the translation module of an NLDBI system needs to secure all possible paraphrases to each domain class or domain class instance. In order to handle this paraphrase problem, our lightweight approach employs both direct and indirect methods, in Section 5.2 and 6.2.1, respectively.

There are two types of translation ambiguities. Class term ambiguity occurs when a class term in a question refers to two or more domain classes. This ambiguity mostly results from general attributes that several domain entities share. Value term ambiguity occurs when a value term corresponds to two or more domain class instances. Date or numeric expressions may almost always cause value term ambiguity. This paper attempts to resolve translation ambiguities using selection restrictions, such as valency information and collocations, in Section 6.2.2.

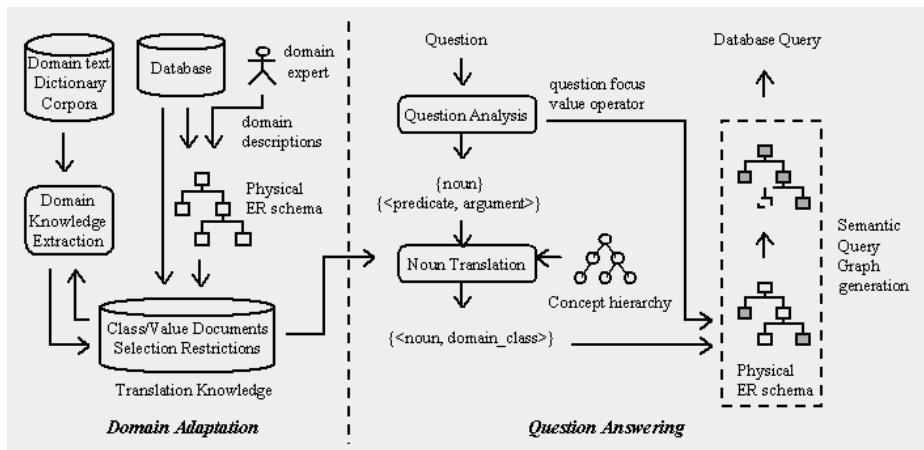


Fig. 1. Lightweight natural language database interfaces (NLDBI) architecture

4 Lightweight NLDBI Architecture

Fig. 1 shows the major steps of a lightweight NLDBI architecture. Currently, this architecture is for relational databases, and uses SQL as a formal database query language. There are two processes: domain adaptation and question answering. For a new database domain, domain adaptation semi-automatically constructs translation knowledge for the subsequent question answering process.

During domain adaptation, initial translation knowledge is first automatically created from both natural language descriptions manually annotated on a physical database schema and entire database values. Then, domain adaptation process continues to automatically adapt the initial translation knowledge into a target database domain using domain-dependent texts, dictionaries, and corpora.

In a question answering process, a user's natural language question is first syntactically analyzed into a set of nouns and a set of predicate-argument pairs. Question analysis also yields a set of feature-value pairs for each question noun. Next, for each question noun, noun translation finds its corresponding domain class, which is then marked on a physical database schema (Physical ER schema in Fig. 1). Finally, from the marked domain classes and question feature-value pairs, a formal database query is generated.

Domain adaptation and question answering are exemplified using English sentences in Section 5 and 6, respectively, while the lightweight approach was originally developed for Korean. So, some language dependent parts are omitted.

5 Domain Adaptation

5.1 Initial Translation Knowledge

For example, consider creating an NLDBI system for a course database in Table 1. First, a database modeling tool is used to generate a physical database schema in Fig. 2.

Table 1. Example of tuples for a course database

Physical Database Schema	Examples of Tuples
T1 (T1C1, T1C2, T1C3)	(1999-0011, Richard, 1999), (2001-0027, Tom, 2001)
T2 (T2C1, T2C2, T2C3)	(ST201A, Statistics, Richard), (ST310B, Algorithms, Joan)
T3 (T3C1, T3C2, T3C3, T3C4)	(1999-0011, ST201A, 1999, A), (2001-0027, ST201A, 2003, C)

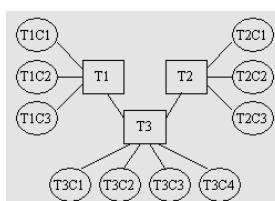


Fig. 2. Physical database schema for a course database

This can be automatically done through a reverse-engineering process that is normally supported by most modern commercial database modeling tools. In Fig. 2, a rectangle indicates a table, a circle refers to a column, and an arc represents either a relationship between tables, or a relationship between a table and its column.

Next, domain experts provide natural language descriptions of a physical database schema according to the following input guidelines. The input process is also graphically supported by database modeling tools.

Input Guidelines	<ul style="list-style-type: none"> ① For each domain class, input its linguistic names in the form of a <i>noun phrase</i> ② For each domain class, input typical domain sentences including related domain classes in the form of a <i>simple sentence</i> ③ In describing domain sentences of ②, a noun phrase referring to a domain class should be either its linguistic name defined in ①, or a domain class itself.
-------------------------	--

Table 2 shows an example of natural language descriptions for the physical database schema in Fig. 2. The reason why we need linguistic annotations is that, on the database side, there are no linguistically homogeneous counterparts that can be related to linguistic terms or concepts used in natural language questions.

Table 2. Natural language descriptions for the physical database schema in Fig. 2.

Domain Class	Natural Language Description	
	Linguistic Name	Domain Sentence
T1	Student	
T1C1	Student identification number	
T1C2	Student name	
T1C3	Entrance year	
T2	Course	
T2C1	Course number	Students take courses in T3C3
T2C2	Course name	Students get grades in T3C3
T2C3	Instructor, professor	Students enter a school in T1C3
T3	Grade	Instructors teach courses
T3C1	Student identification number	Instructors give grades
T3C2	Course number	Courses are open in T3C3
T3C3	Grade year	
T3C4	Grade	

A physical database schema annotated with natural language descriptions is called a *pER* (physically-derived Entity-Relationship) *schema*. The term *pER schema* was coined because we believe that it is an approximation of the target database's original ER (Entity-Relationship) schema, in the sense that each component in *pER schema* has a conceptual-level label (natural language description), and its structures are directly derived from a physical database schema. Thus, *pER schema* has the potential to bridge between linguistic constructions and physical database structures.

From *pER schema*, translation knowledge is created. We define translation knowledge into two structures: *class-referring* and *class-constraining*. Class-referring

translation knowledge falls into two kinds: class documents and value documents. A **class document** contains a set of class terms referring to a domain class. Class terms are extracted from linguistic names for domain classes within the *pER schema*, by generating head-preserving noun phrases, as shown in Table 3. A **value document** is created from a set of domain class instances associated with a column.

As class-constraining translation knowledge, there are two types of selection restrictions: valency-based, and collocation-based. In this paper, **valency-based** selection restriction describes a set of domain classes of arguments that verbs or prepositions take, as shown in Table 4. A **Collocation-based** one is simply a collection of collocations of each domain class in the form of collocation documents, as shown in Table 5. Therefore, given n tables and m columns per table, $n+3nm$ documents are created. In other words, for each column, three documents are created: class, value, and collocation documents.

Then, in our case, except for valency-based selection restriction, constructing translation knowledge corresponds to indexing documents related to domain classes. Also, translating a question noun into its domain classes means retrieving relevant documents using a question noun as an IR query. This will be explained in Section 6.

Table 3, 4, and 5 show an example of translation knowledge associated with Table 1 and 2. Class documents and selection restrictions can be automatically extracted, because linguistic descriptions of *pER schema* are subject to input guidelines that restrict natural language expressions into simple sentences and a closed set of linguistic names and domain classes. Value documents are created using value patterns that are automatically determined by pattern-based n-gram indexing [7].

Table 3. Class-referring translation knowledge

Do- main Class	Class-Referring Translation Knowledge	
	Class Document	Value Document
T1	Student	NULL
T1C1	Student identification number, identification number, number	$n4s1n4, n4, s1, n4s1, s1n4$
T1C2	Student name, name	Richard, Tom
T1C3	Entrance year, year	1999, 2003
T2	Course	NULL
T2C1	Course number, number	$c2n3c1, c2, n3, c1, c2n3, n3c1$
T2C2	Course name, name	Statistics, Algorithms
...
T3C4	Grade	A, C

In Table 3, $n4s1n4$ is a value pattern of a domain class T1C1. $n4s1n4$ means that values (e.g., 1999-0011 or 2001-0027) of T1C1 typically start with 4 decimal numbers ($n4$) followed by 1 special character ($s1$), and end with 4 decimal numbers ($n4$). A value pattern $c2n3c1$ of T2C1 can be explained similarly. In this paper, $n4s1n4$, or $c2n3c1$ are called canonical patterns. This pattern-based value indexing [4] reduces an open-ended set of alphanumeric terms into a closed set of patterns. However, the method [4] cannot deal with random alphanumeric value terms that are not captured by predefined patterns, and only support exact matching.

To overcome the limitations of pattern-based value indexing [4], we further generate 1-grams and 2-grams in a pattern unit from canonical patterns. For example, $n4s1n4$ generates $n4$, $s1$ as 1-grams, and $n4s1$, $s1n4$ as 2-grams. These pattern-based n-grams [7] enable partial matching between patterns, and also may reduce indexing storage over storing canonical patterns, since canonical patterns are sliced into smaller n-grams that will have many duplicate n-grams. For example, given a question *Show me courses starting with ‘st’*, we can map ‘st’ into T2C1, through an n-gram pattern $c2$ in a value document of T2C1. These pattern-based n-grams [7] are one way to handle value term paraphrases referred to in Section 3.

Table 4. Class-constraining translation knowledge (Valency-based selection restriction)

Predicate or Preposition	Set of Domain Classes
Take	T1, T2, T3C3
Get	T1, T3, T3C4, T3C3
Enter	T1, T1C3
Teach	T2C3, T2
Give	T2C3, T3, T3C4
Open	T2, T3C3
In	T1C3, T3C3

In Table 4, a set of domain classes is obtained by merging domain classes of arguments that are governed by the same predicate or preposition in domain sentences of Table 2.

Table 5. Class-constraining translation knowledge (collocation-based selection restriction)

Domain Class	Collocation Document
T1	NULL
T1C1	Student, identification
T1C2	Student
T1C3	Student, Entrance
T2	NULL
T2C1	Course
...	...

In Table 5, collocation words of each domain class are acquired from its linguistic names in the *pER schema*, by gathering all the other words except the rightmost head of each linguistic name. In addition, for each domain class corresponding to a column, its collocation document additionally includes all terms in the class document of a table to which that column belongs. For example, in the collocation document for T1C3, ‘student’ was inserted from the class document of T1.

5.2 Expansion of Initial Translation Knowledge

To directly attack class term paraphrases, the initial translation knowledge may be further extended by domain knowledge extraction from domain materials, dictionaries, or corpora. Domain materials are difficult to obtain electronically. So, currently,

our method extracts domain knowledge (domain-dependent translation knowledge) from dictionaries and corpora. In this paper, we describe translation knowledge expansion using only dictionaries.

5.2.1 Dictionary

Table 6 shows an example of an extended class document for a domain class T2C3 from WordNet [10]. Extension procedures using WordNet are as follows. First, for each class term in an initial class document, we obtain its genus terms from its definitions in WordNet. For example, *person* italicized in Table 6 is a genus term of a class term *instructor*. Then, using the WordNet taxonomic hierarchy, all hypernyms of *instructor* are gathered until a genus term *person* or a taxonomic root is met. The extracted hypernyms are inserted into the initial class document of T2C3. In the case of *professor*, its hypernyms are similarly extracted using the genus term *someone*. Superscripts in Table 6 indicate hypernym levels from its initial class term in the taxonomic hierarchy. The superscript is used to discriminate class terms in terms of its class-referring strength, since a larger superscript means a more generalized class term. Superscript 0 means synonyms that are obtained in WordNet from the synset node of each initial class term.

Table 6. Example of an extended class document

Domain Class	Class Document	
	Initial	Extended
T2C3	Instructor, Professor	Instructor, teacher ⁰ , educator ¹ , pedagogue ¹ , professional ² , professional_person ² , adult ³ , grownup ³ , <i>person</i> ⁴ , individual ⁴ , someone ⁴ , somebody ⁴ , mortal ⁴ , human ⁴ , soul ⁴
		Professor, academician ¹ , academic ¹ , faculty_member ¹ , educator ² , pedagogue ² , professional ³ , professional_person ³ , adult ⁴ , grownup ⁴ , person ⁵ , individual ⁵ , someone ⁵ , somebody ⁵ , mortal ⁵ , human ⁵ , soul ⁵

However, an initial class term may have several senses in dictionaries. So, we need a certain word sense disambiguation (WSD) technique to eliminate inclusion of noisy hypernyms from incorrect senses. For this, we employ a context vector approach [11]. First, a target class term t is converted into a context vector V_t that is a set of all words except t in entire natural language descriptions of *pER schema*. Next, for each dictionary sense s of t , we create a context vector V_s that is a set of all words except t in a definition sentence of s . Then, we disambiguate t based on Formula 1 that measures the cosine similarity between two vectors.

$$s^* = \arg \max_s \frac{V_s \bullet V_t}{|V_s| \times |V_t|} \quad (1)$$

Note that if a class document D corresponding to a table is changed through this domain knowledge extraction, a collocation document for each column that belongs to the table is merged with D to become a larger one.

6 Question Answering

6.1 Question Analysis

The question answering proceeds as follows. A user writes his or her information need in a natural language. Next, the user question is analyzed to produce a set of question nouns and a set of predicate-argument pairs. In Korean, these can be obtained after morphological analysis, tagging, chunking, and dependency parsing. Question analysis also yields a set of feature-value pairs for each question noun. Among others, important question features are a question focus and a value operator, because these two features determine select-clause items and where-clause operators, respectively, in a final SQL query.

For example, consider a user question “*show me the name of students who got A in statistics from 1999*”. After question analysis, we obtain the first four columns in Table 7. Actually, ‘*show*’ is the governor of ‘*name*’, but we use ‘*show*’ only to determine which question nouns are question foci.

Table 7. Analysis of an example question

Question Noun	Head Verb	Question Features		Relevant Domain Classes	Disambiguated Domain Classes
		Question Focus	Value Operator		
Name		Yes		T1C2 ^c , T2C2 ^c	T1C2 ^c
Student	Get	No	=	T1 ^c	T1 ^c
A	Get	No	=	T3C4 ^v	T3C4 ^v
Statistics	Get	No	=	T2C2 ^v	T2C2 ^v
1999	Get	No	>=	T1C3 ^v , T3C3 ^v	T3C3 ^v

6.2 Noun Translation

Noun translation utilizes an IR framework to translate each question noun into a probable domain class. First, **class retrieval** converts each question noun into an IR query, and retrieves relevant documents from class-referring translation knowledge; that is, a collection of entire class documents and value documents. Here, retrieved documents mean the candidate domain classes for the question noun, because each document is associated with a domain class. Next, **class disambiguation** selects a likely domain class among the candidate domain classes retrieved by class retrieval, using class-constraining translation knowledge.

6.2.1 Class Retrieval

For each question noun, class retrieval retrieves lexically or semantically equivalent domain classes in the form of class or value documents using an IR framework, where the question noun is transformed into an IR query. In Table 7, superscript *c* or *v* means that the associated domain class are obtained by retrieving class documents or value documents, respectively. For example, ‘A’ was mapped into ‘T3C4^v’ by retrieving a value document of T3C4, and ‘*name*’ was mapped into ‘T1C2^c’ or

‘T2C2^c’ by retrieving a class document of T1C2 or T2C2. This distinction between class and value documents is important, because if a question noun retrieves a value document, it needs to participate in an SQL-where condition.

The reason why an IR method is employed is twofold. First, the translation knowledge construction process is simplified to document indexing. Thus, translation knowledge is easily scalable to a larger size. Thus far, translation knowledge construction requires human expertise, such as AI/NLP/DBMS, so its construction process was time-consuming and tedious. For example, in modern logical form systems, a logical predicate should be defined for each lexical item, then a database relation is to be defined for the logical predicate.

Second, IR can support a variety of matching schemes (e.g., lexical or conceptual matching, exact or approximate matching, and word or phrase matching), which are required to relate various forms of paraphrases for domain classes or domain class instances. These various IR matching techniques correspond to indirect methods in order to handle class term or value term paraphrase problems.

6.2.2 Class Disambiguation

After class retrieval, two types of translation ambiguities introduced in Section 3 may occur. In our example question, class term ambiguity occurs on a question noun ‘name’, and value term ambiguity on ‘1999’. Then, to each question noun holding two or more relevant domain classes, selection restrictions in Table 4 and 5 are applied to determine the correct domain classes.

In order to resolve the value term ambiguity of ‘1999’, ‘get’ (a governor of ‘1999’) is lexically or conceptually searched over valency-based selection restrictions in Table 4 to find that a domain verb ‘get’ requires its arguments to be $B=\{T1, T3, T3C4, T3C3\}$. Let A be a set of relevant domain classes of ‘1999’ ($A=\{T1C3, T3C3\}$). Then, $A \cap B$ is calculated to disambiguate ‘1999’ into a correct domain class T3C3.

However, valency-based selection restrictions in Table 4 cannot be applied to translation ambiguity whose target word does not have predicates or prepositions as its governor, like ‘name’. In this case, collocation-based selection restrictions in Table 5 are used as follows. To disambiguate ‘name’, an adjacent word ‘student’ of a target word ‘name’ is used as an IR query to retrieve relevant collocation documents $B=\{T1C1, T1C2, T1C3\}$. Note that elements in B are not class documents but collocation document. Let A be a set of relevant domain classes of ‘name’ ($A=\{T1C2, T2C2\}$). Then, similarly, $A \cap B$ is calculated to disambiguate ‘name’ into a correct domain class T1C2.

6.3 Database Query Generation

In this stage, if any ambiguous question noun occurs having two or more domain classes, a user confirmation feedback procedure is fired in order to ask a user to indicate a correct domain class. Otherwise, the following query generation procedure begins. For query generation, we adopt the Zhang’s method [15].

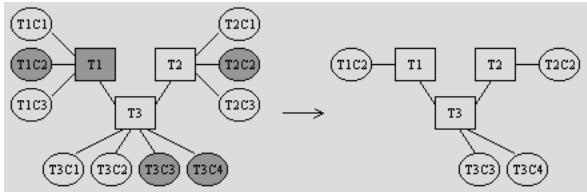


Fig. 3. Finding a query graph from a physical database schema

Disambiguated domain classes constitute a sub-graph on the physical database schema, which is viewed as a graph where a node corresponds to a table or a column, and an arc is defined by a relationship between two tables, or a property between a table and its column. Fig. 3 shows the sub-graph that we call a query graph. In our case, the problem is to locate a sub-graph with a minimal number of nodes.

From the query graph and question features in Table 7, a formal database query is generated. The entity nodes of a query graph are transformed to an SQL-from clause, and arcs between entities constitute SQL join operators in an SQL-where clause. An SQL-select clause is obtained from question focus features, and value operator features are applied to the corresponding attribute nodes to produce an SQL-where clause. Note that value operators are fired only on the question nouns that retrieved value documents. Thus, the final SQL query is as follows.

```

SELECT  T1C2
FROM    T1, T2, T3
WHERE   T1.T1C1 = T3.T3C1 AND T2.T2C1 = T3.T3C2
        AND T2C2 = 'Statistics' AND T3C3 = 'A' AND T3C4 >= 1999

```

7 Discussions and Conclusion

To alleviate the transportability problem of NLDBI translation knowledge, this paper presented an overview of a lightweight NLDBI approach based on an IR framework, where most translation knowledge is simplified in the form of documents such as class, value, and collocation documents. Thus, translation knowledge construction is reduced to document indexing, and noun translation can be carried out based on document retrieval. In addition, translation knowledge can be easily expanded from other resources. Another motivation is to create a robust NLDBI system. For the robustness, we used a syntactic analyzer only to obtain question nouns and its predicate-argument pairs, rather than entire parse trees.

Meng et al. [9] proposed a similar IR approach. However, our approach mainly differs from theirs in terms of the following points. First, we focused on automated construction and its scalable expansion of translation knowledge, in order to increase transportability. Next, our disambiguation strategy in noun translation relies on linguistically motivated selection restrictions that are extracted from predicate-argument pairs of domain predicates. However, Meng et al. [9] used neighboring words as disambiguation constraints, because their method does not perform any syntactic analysis.

Currently, the lightweight approach was applied to only one database domain [7]. In future, it requires much empirical evaluation to validate its operability. For example, input guidelines in Section 5.1 should be tested on both different user groups and several database domains. In addition, we assumed that question meaning is approximated by a query graph (and its structural constraints) on a physical database schema, that is determined using question nouns and its predicate-argument structures. So, the lightweight method may implicitly restrict expressiveness of user questions. We plan to investigate how many and what kinds of user questions are well suited to the lightweight approach.

Acknowledgements. This work was supported by the KOSEF through the Advanced Information Technology Research Center (AITrc) and by the BK21 project. We sincerely thank the anonymous reviewers for their invaluable suggestions.

References

1. Androutsopoulos, I.: Interfacing a Natural Language Front-End to Relational Database. *Master's thesis*, Technical Report 11, Department of Artificial Intelligence, University of Edinburgh (1993)
2. Androutsopoulos, I., Ritchie, G.D., and Thanisch, P.: Natural Language Interfaces to Databases – An Introduction. *Natural Language Engineering* 1(1) (1995) 29-81
3. Ballard, B.W., Lusth, J.C., and Tinkham, N.L.: LDC-1: A Transportable, Knowledge-Based Natural Language Processor for Office Environments. *ACM Transactions on Office Information Systems* 2(1) (1984) 1-25
4. Damerau, F.: Problems and Some Solutions in Customization of Natural Language Database Front Ends. *ACM Transactions on Office Information Systems* 3(2) (1985) 165-184
5. Grosz, B.J., Appelt, D.E., Martin, P.A., and Pereira, F.C.N.: TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces. *Artificial Intelligence* 32(2) (1987) 173-243
6. Hendrix, G.G., Sacerdoti, D., Sagalowicz, D., and Slocum, J.: Developing a Natural Language Interface to Complex Data. *ACM Transactions on Database Systems* 3(2) (1978) 105-147
7. Kang, I.S., Bae, J.H., Lee, J.H.: Natural Language Database Access using Semi-Automatically Constructed Translation Knowledge. *Proceedings of the 1st International Joint Conference on Natural Language Processing*, Hainan, China, (2004) 512-519
8. Lee, H.D., and Park, J.C.: Interpretation of Natural language Queries for Relational Database Access with Combinatory Categorial Grammar. *International Journal of Computer Processing of Oriental Languages* 15(3) (2002) 281-304
9. Meng, F., and Chu, W.W.: Database Query Formation from Natural Language using Semantic Modeling and Statistical Keyword Meaning Disambiguation. *Technical Report*, CSD-TR 990003, University of California, Los Angeles (1999)
10. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K.: Five Papers on WordNet. *Special Issue of International Journal of Lexicography* 3(4) (1990) 235-312
11. Schutze, H.: Context Space. *Working Notes of AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, Cambridge, MA (1992) 113-120

12. Templeton, M., and Burger, J.: Problems in Natural Language Interface to DBMS with Examples with EUFID. *Proceeding of the 1st Conference on Applied Natural Language Processing*, Santa Monica, California (1983) 3-16
13. Waltz, D.L.: An English Language Question Answering System for a Large Relational Database. *Communications of the ACM* 21(7) (1978) 526-539
14. Warren, D., and Pereira, F.: An Efficient Easily Adaptable System for Interpreting Natural Language Queries. *Computational Linguistics* 8(3-4) (1982) 110-122
15. Zhang, G., Chu, W.W., Meng, F., and Kong, G.: Query Formulation from High-level Concepts for Relational Databases. *Proceedings of the 1st International Workshop on User Interfaces to Data Intensive Systems*, Edinburgh, Scotland, (1999) 64-75

Ontology-Driven Question Answering in AquaLog

Vanessa Lopez and Enrico Motta

Knowledge Media Institute, The Open University.
Walton Hall, Milton Keynes,
MK7 6AA, United Kingdom.
{v.lopez, e.motta}@open.ac.uk

Abstract The *semantic web* vision is one in which rich, *ontology-based semantic markup* is widely available, both to enable sophisticated interoperability among agents and to support human web users in locating and making sense of information. The availability of semantic markup on the web also opens the way to novel, sophisticated forms of question answering. AquaLog is a portable question-answering system which takes queries expressed in natural language and an ontology as input and returns answers drawn from one or more knowledge bases (KBs), which instantiate the input ontology with domain-specific information. AquaLog makes use of the GATE NLP platform, string metrics algorithms, WordNet and a novel ontology-based *relation similarity service* to make sense of user queries with respect to the target knowledge base. Finally, although AquaLog has primarily been designed for use with semantic web languages, it makes use of a generic plug-in mechanism, which means it can be easily interfaced to different ontology servers and knowledge representation platforms.

1 Introduction

The *semantic web* vision [1] is one in which rich, *ontology-based semantic markup* is widely available, both to enable sophisticated interoperability among agents, e.g., in the e-commerce area, and to support human web users in locating and making sense of information. For instance, tools such as Magpie [2] support sense-making and semantic web browsing by allowing users to select a particular ontology and use it as a kind of ‘semantic lens’, which assists them in making sense of the information they are looking at. As discussed by McGuinness in her recent essay on “Question Answering on the Semantic Web” [3], the availability of semantic markup on the web also opens the way to novel, sophisticated forms of question answering, which not only can potentially provide increased precision and recall compared to today’s search engines, but are also capable of offering additional functionalities, such as i) proactively offering additional information about an answer, ii) providing measures of reliability and trust and/or iii) explaining how the answer was derived.

While semantic information can be used in several different ways to improve question answering, an important (and fairly obvious) consequence of the availability

of semantic markup on the web is that this can indeed be queried directly. For instance, we are currently augmenting our departmental web site, <http://kmi.open.ac.uk>, with semantic markup, by instantiating an ontology describing academic life [4] with information about our personnel, projects, technologies, events, etc., which is automatically extracted from departmental databases and unstructured web pages. In the context of standard, keyword-based search this semantic markup makes it possible to ensure that standard search queries, such as “peter scott home page kmi”, actually return Dr Peter Scott’s home page as their first result, rather than some other resource (as indeed is the case when using current non-semantic search engines on this particular query). Moreover, as pointed out above, we can also query this semantic markup directly. For instance, we can ask a query such as “list all the kmi projects in the semantic web area” and, thanks to an inference engine able to reason about the semantic markup and draw inferences from axioms in the ontology, we can then get the correct answer.

This scenario is of course very similar to asking natural language queries to databases (NLDB), which has long been an area of research in the artificial intelligence and database communities [5] [6] [7] [8] [9], even if in the past decade has somewhat gone out of fashion [10] [11]. However, it is our view that the semantic web provides a new and potentially very important context in which results from this area of research can be applied. Moreover, interestingly from a research point of view, it provides a new ‘twist’ on the old issues associated with NLDB research. Hence, in the first instance, the work on the AquaLog query answering system described in this paper is based on the premise that the semantic web will benefit from the availability of natural language query interfaces, which allow users to query semantic markup viewed as a knowledge base. Moreover, similarly to the approach we have adopted in the Magpie system, we believe that in the semantic web scenario it makes sense to provide query answering systems on the semantic web, *which are portable with respect to ontologies*. In other words, just like in the case of Magpie, where the user is able to select an ontology (essentially a semantic viewpoint) and then browse the web through this semantic filter, our AquaLog system allows the user to choose an ontology and then ask queries with respect to the universe of discourse covered by the ontology.

2 The AquaLog Approach

AquaLog is a portable question-answering system which takes queries expressed in natural language and an ontology as input and returns answers drawn from one or more knowledge bases (KBs), which instantiate the input ontology with domain-specific information. As already emphasized, a key feature of AquaLog is that it is modular with respect to the input ontology, the aim here being that it should be zero cost to switch from one ontology to another when using AquaLog.

AquaLog is part of the AQUA [12] framework for question answering on the semantic web and in particular addresses the upstream part of the AQUA process, the

translation of NL queries into logical ones and the interpretation of these NL-derived logical queries with respect to a given ontology and available semantic markup.

AquaLog adopts a *triple-based* data model and, as shown in figure 1, the role of the linguistic component of AquaLog is to translate the input query into a set of intermediate triples, of the form <subject, predicate, object>. These are then further processed by a module called the “Relation Similarity Service” (RSS), to produce ontology-compliant queries. For example, in the context of the academic domain mentioned earlier, AquaLog is able to translate the question “Who is a Professor at the Knowledge Media Institute?” into the following, ontology-compliant logical query, <typeOf ?x Professor-in-Academia> & <works-in-unit ?x KMi>, expressed as a conjunction of non-ground triples (i.e., triples containing variables). The role of the RSS is to map the intermediate form, <?who, Professor, KMi> into the target, ontology-compliant query.

There are two main reasons for adopting a triple-based data model. First of all, as Katz et al. point out [13], although not all possible queries can be represented in the binary relational model, in practice these occur very frequently. Secondly, RDF-based knowledge representation (KR) formalisms for the semantic web, such as RDF itself [14] or OWL [15] also subscribe to this binary relational model and express statements as <subject, predicate, object>. Hence, it makes sense for a query system targeted at the semantic web to adopt this data model.

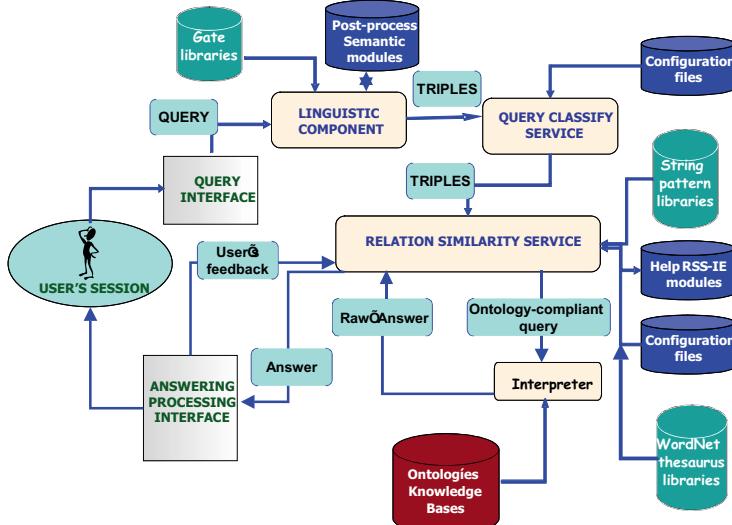


Fig. 1. The Architecture of AquaLog

We have seen that, in common with most other NLDB systems, AquaLog divides the task of mapping user queries to answers into two main subtasks: producing an intermediate logical representation from the input query and mapping this intermediate query into a form consistent with the target knowledge base. Moreover it explic-

itly restricts the range of questions the user is allowed to ask to a set of expressions/syntactic patterns, so that the linguistic limits of the system are obvious to the user (to avoid the effort of rephrasing questions) and to ensure users understand whether a query to AquaLog failed for reasons which are *linguistic* (failure to understand the linguistic structure of the question) or *conceptual* (failure of the ontology to provide meaning to the concepts in the query).

In the next section we describe the AquaLog architecture in more detail.

3 The AquaLog Architecture

AquaLog was designed with the aim of making the system as flexible and as modular as possible. It is implemented in Java as a web application, using a client-server architecture. A key feature is the use of a plug-in mechanism, which allows AquaLog to be configured for different KR languages. Currently we use it with our own OCML-based KR infrastructure [16] [17], although in the future we plan to provide direct plug-in mechanisms for use with the emerging RDF and OWL servers¹.

3.1 Initialization and User's Session

At initialization time the AquaLog server will access and cache basic indexing data for the target KB(s), so that they can be efficiently accessed by the remote clients to guarantee real-time question answering, even when multiple users access the server simultaneously.

3.2 Gate Framework for Natural Language and Query Classify Service

AquaLog uses the GATE [18] [19] infrastructure and resources (language resources, processing resources like ANNIE, serial controllers, etc.) to map the input query in natural language to the triple-based data model. Communication between AquaLog and GATE takes place through the standard GATE API. The GATE chunker used for this task does not actually generate a parse tree. As discussed by Katz et al. [20] [21], although parse trees (as for example, the NLP parser for Stanford [22]) capture syntactic relations, they are often time-consuming and difficult to manipulate. Moreover, we also found that in our context we could do away with much of the parsed information. For the intermediate representation, we use the triple-based data model rather than logic, because at this stage we do not have to worry about getting the representation right. The role of the intermediate representation is simply to provide an easy to manipulate input for the RSS.

¹ In any case we are able to import and export RDF(S) and OWL from OCML, so the lack of an explicit RDF/OWL plug-in is actually not a problem in practice.

After the execution of the GATE controller a set of syntactical annotations are returned associated with the input query. Annotations include information about sentences, tokens, nouns and verbs. For example we get voice and tense for the verbs, or categories for the nouns, such as determinant, singular/plural, conjunction, possessive, determiner, preposition, existential, wh-determiner, etc. When developing AquaLog we extended the set of annotations returned by GATE, by identifying relations and question indicators (which/who/when/etc.). This was achieved through the use of *Jape grammars*. These consist of a set of *phases*, which run sequentially, each of which defined as a set of pattern rules. The reason for this extension was to be able to clearly identify the scope of a relation – e.g., to be able to identify “has research interest” as a relation. Here we exploited the fact that natural language commonly employs a preposition to express a relationship.

Although it is not necessary, we could improve the precision of the AquaLog linguistic component by modifying or creating the appropriate jape rules for specific cases. For example, the word “project” could be understood as a noun or as a verb, depending on the priority of the rules. Another example is when some disambiguation is necessary as in the example: “Has john research interest in ontologies?”. Here “research” could be either the last name of John or a noun part of the relation “has research interest”².

As shown in figure 1, the architecture also includes a post-processing semantic (PPS) module to further process the annotations obtained from the extended GATE component. For example when processing the query “John has research interest in ontologies”, the PPS ensures that the relation is identified as “has research interest”. Other more complex cases are also dealt with.

Finally, before passing the intermediate triples to the RSS, AquaLog performs two additional checks. If it is not possible to transform the question into a term-relation form or the question is not recognized, further explanation is given to the user, to help him to reformulate the question. If the question is valid, then a Query Classify Service is invoked to determine the type of the question, e.g., a “where” question, and pass this information on to the Relation Similarity Service.

3.3 The Relation Similarity Service

This is the backbone of the question-answering system. The RSS is called after the NL query has been transformed into a term-relation form and classified and it is the main component responsible for producing an ontology-compliant logical query.

Essentially the RSS tries to make sense of the input query by looking at the structure of the ontology and the information stored in the target KBs, as well as using string similarity matching and lexical resources, such as WordNet. There is not a single strategy here, so we will not attempt to give a comprehensive overview of all

² Of course a better way to express the query would be “Has John got a research interest in ontologies?”, which can be parsed with no problems. However, in our experience these slightly un-grammatical queries are very common and it is our aim to produce a system robust enough to deal with many of them.

the possible ways in which a query can be interpreted. Rather, we will show one example, which is illustrative of the way the RSS works.

An important aspect of the RSS is that it is interactive. In other words when unsure it will ask the user for help, e.g., when it is not able to disambiguate between two possible relations which can be used to interpret the query.

For example, let's consider a simple question like “*who works on the semantic web?*”. Here, the first step for the RSS is to identify that “semantic web” is actually a “research area” in the target KB³. If a successful match is found, the problem becomes one of finding a relation which links a person (or an organization) to the semantic web area.



Fig. 2. AquaLog in action.

By analyzing the KB, AquaLog finds that the only relation between a person and the semantic web area is has-research-interest and therefore suggests to the user that the question could be interpreted in terms of this relation. If the user clicks OK, then the answer to the query is provided. It is important to note that in order to make sense of the triple <person, works, semantic web>, all subclasses of person need to be considered, given that the relation has-research-interest could be defined only for researchers rather than people in general. If multiple relations are possible candidates for interpreting the query, then string matching is used to determine the most likely candidate, using the relation name, eventual aliases, or synonyms provided by lexical resources such as WordNet [23]. If no relations are found using this method, then the

³ Naming convention vary depending on the KR used to represent the KB and may even change with ontologies – e.g., an ontology can have slots such as “variant name” or “pretty name”. AquaLog deals with differences between KRs by means of the plug-in mechanism. Differences between ontologies need to be handled by specifying this information in a configuration file.

user is asked to choose from the current list of candidates. However, it is important to emphasise that calling on the users to disambiguate is only done if no information is available to AquaLog, which allows the system to disambiguate the query directly. For instance let's consider the two queries shown in figure 3. On the right screen we are looking for the web address of Peter and given that the system is unable to disambiguate between Peter-Scott, Peter-Sharpe or Peter-Whalley, user's feedback is required. However, on the left screen we are asking for the web address of Peter, who has an interest in knowledge reuse. In this case AquaLog does not need assistance from the user, given that only one of the three Peters has an interest in knowledge reuse.

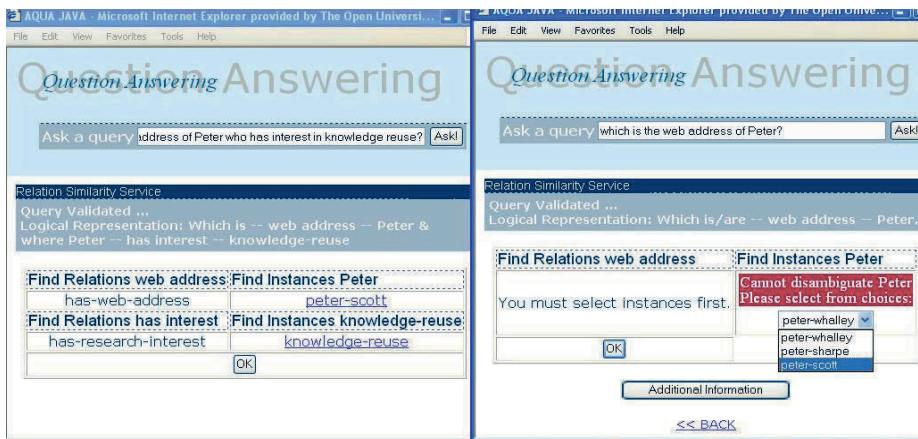


Fig. 3. Automatic or user-driven disambiguation.

A typical situation the RSS has to cope with is one in which the structure of the intermediate query does not match the way the information is represented in the ontology. For instance, the query “which are the publications of John?” may be parsed into <John, publications, something>, while the ontology may be organized in terms of <Publication, has-author, Author>. Also in these cases the RSS is able to reason about the mismatch and generate the correct logical query.

3.4 Helping the User Making Sense of the Information

We have already mentioned that AquaLog is an interactive system. If the RSS fails to make sense of a query, the user is asked to help choose the right relation or instance. Moreover, in order to help the user, AquaLog shows as much information as possible about the current query, including both information drawn from WordNet and from the ontology – see figure 4.

Left Screenshot (Question Answering):

Ask a query: what researchers are managed by Motta?

WordNet Synonyms for "managed":	Relations for:	Possible relations for:
pull_off negociate bring_off carry_off manage deal care	researcher	researcher in which the type is senior-research-fellow-in-academia person-being-visited (employee) is-manager-of (person)

Right Screenshot (Relation Similarity Service):

WordNet Synonyms for "managed":	Relations for:	Possible relations for:
pull_off negociate bring_off carry_off manage deal care	researcher	researcher in which the type is senior-research-fellow-in-academia person-being-visited (employee) is-manager-of (person)
	senior-research-fellow-in-academia	senior-research-fellow-in-academia in which the type is researcher is-manager-of (person)

Fig. 4. Providing additional information

Finally, when answers are presented to the user, it is also possible for him/her to use these answers as starting points for navigating the KB – see figure 5.

Left Screenshot (Question Answering):

Ask a query: what researchers are managed by Motta?

Relation Similar Service

Query Validated ...
Logical Representation: researcher -- is-manager-of -- enrico-motta.

The answer to the question:

john-domingue	zdenek-zdrahal	arthur-stutt
peter-whalley	maria-vargas-vera	vanessa-lopez
jianhan-zhu	fershad-hakimpour	

Right Screenshot (Instance Information):

Instance	vanessa-lopez
Home ontology	aktive-portal-kb
Instance of	research-fellow-in-academia
has-variant-name	
has-time-interval	
has-web-address	
has-email-address	'v.lopez@open.ac.uk'
has-telephone-number	
has-fax-number	
has-postal-address	, km1-postal-address
full-name	, "vanessa lopez"
family-name	, "lopez"
given-name	, "vanessa"
has-gender	
has-academic-degree	
has-appellation	
has-id	, 110
has-outcu	, v1474"
Is-manager-of	
has-work-status	
works-in-unit	, knowledge-media-institute-at-the-open-university
has-job-title	, "research fellow"
has-contract-type	
works-for	, the-open-university

Fig. 5. Navigating the KB starting from answers to queries.

3.5 String Matching Algorithms

String algorithms are used to find patterns in the ontology for any of the terms inside the intermediate triples obtained from the user's query. They are based on String Distance Metrics for Name-Matching Tasks, using an open-source from the Carnegie Mellon University in Pittsburgh [24]. This comprises a number of string distance metrics proposed by different communities, including edit-distance metrics, fast heuristic string comparators, token-based distance metrics, and hybrid methods. The conclusions of an experimental comparison of metrics realized by the University of Pittsburgh states: "Overall, the best-performing method is an hybrid scheme combining a TFIDF weighting scheme ... with the Jaro-Winkler string distance scheme, developed in the probabilistic record linkage community".

However, it is not recommendable to trust just one metric. Experimental comparisons using the AKT ontology with different metrics show that the best performance is obtained with a combination of the following metrics: JaroWinkler, Level2JaroWinkler and Jaro.

A key aspect of using metrics is *thresholds*. By default two kinds of thresholds are used, called: "trustable" and "did you mean?", where the former is of course always preferable to the latter. AquaLog uses different thresholds depending on whether it is looking for concepts names, relations or instances.

4 Evaluation

4.1 Scenario and Results

A full evaluation of AquaLog will require both an evaluation of its query answering ability as well an evaluation of the overall user experience. Moreover, because one of our key aims is to make AquaLog portable across ontologies, this aspect will also have to be evaluated formally. While a full evaluation has not been carried out yet, we performed an initial study, whose aim was to assess to what extent the AquaLog application built using AquaLog with the AKT ontology and the KMi knowledge base satisfied user expectations about the range of questions the system should be able to answer. A second aim of the experiment was also to provide information about the nature of the possible extensions needed to the ontology and the linguistic components – i.e., not only we wanted to assess the current coverage of the system but also get some data about the complexity of the possible changes required to generate the next version of the system. Thus, we asked ten members of KMi, none of whom has been involved in the AquaLog project, to generate questions for the system. Because one of the aims of the experiment was to measure the linguistic coverage of the system with respect to user needs, we did not provide them with any information about the linguistic ability of the system. However, we did tell them something about the conceptual coverage of the ontology, pointing out that its aim was to model the key elements of a research lab, such as publications, technologies, projects, research areas, people, etc.

We also pointed out that the current KB is limited in its handling of temporal information, therefore we asked them not to ask questions which required sophisticated temporal reasoning. Because no ‘quality control’ was carried out on the questions, it was perfectly OK for these to contain spelling mistakes and even grammatical errors.

We collected in total 76 different questions, 37 of which were handled correctly by AquaLog, i.e., 48.68% of the total. This was a pretty good result, considering that no linguistic restriction was imposed on the questions.

We analyzed the failures and divided them into the following five categories (the total adds up to more than 37 because a query may fail at several different levels):

- **Linguistic failure.** This occurs when the NLP component is unable to generate the intermediate representation (but the question can usually be reformulated and answered). This was by far the most common problem, occurring in 27 of the 39 queries not handled by AquaLog (69%).
- **Data model failure.** This occurs when the NL query is simply too complicated for the intermediate representation. Intriguingly this type of failure never occurred, and our intuition is that this was the case not only because the relational model is actually a pretty good way to model queries but also because the ontology-driven nature of the exercise ensured that people only formulated queries that could in theory (if not in practice) be answered by reasoning about the departmental KB.
- **RSS failure.** This occurs when the relation similarity service of AquaLog is unable to map an intermediate representation to the correct ontology-compliant logical expression. Only 3 of the 39 queries not handled by AquaLog (7.6%) fall into this category.
- **Conceptual failure.** This occurs when the ontology does not cover the query. Only 4 of the 39 queries not handled by AquaLog (10.2%) failed for this reason.
- **Service failure.** Several queries essentially asked for services to be defined over the ontologies. For instance one query asked about “the top researchers”, which requires a mechanism for ranking researchers in the lab - people could be ranked according to citation impact, formal status in the department, etc. In the context of the semantic web, we believe that these failures are less to do with shortcomings of the ontology than with the lack of appropriate services, defined over the ontology. Therefore we defined this additional category which accounted for 8 of the 39 failures (20.5%).

4.2 Discussion

Here we briefly discuss the issues raised by the evaluation study and in particular what can be done to improve the performance of the AquaLog system.

Service failures can of course be solved by implementing the appropriate services. Some of these can actually be to some extent ontology-independent, such as “similarity services”, which can answer questions like “Is there a project similar to AKT?”. Other services can be generically categorized (e.g., “ranking services”), but will have

to be defined for specific concepts in an ontology, such as mechanisms to rank people publications, or projects. Here we envisage a solution similar to the one used in the Magpie tool [2], where service developers are given publishing rights to develop and associate services to concepts in an ontology, using semantic web service platforms such as IRS-II [25].

The few RSS failures basically highlighted bugs in AquaLog all of which can be fixed quite easily. A clear example of this is the query “who funds the magpie project”, where “who” is understood to be a person, while it of course can also be an organization or funding body.

The few conceptual failures are also easy to fix, they highlighted omissions in the ontology.

The most common and problematic errors are linguistic ones, which occurred for a number of reasons:

In some cases people asked new types of basic queries outside the current linguistic coverage, formed with: “how long”, “there are/is”, “are/is there any”, “how many”, etc.

Some problems were due to combinations of basic queries, such as “What are the publications in KMi related to social aspects in collaboration and learning?”, which the NLP component of AquaLog cannot currently untangle correctly. Another example is when one of the terms is “hidden” because it is included in the relation but actually it is not part of the relation, as for example in “Who are the partners involved in the AKT project?” One may think the relation is “partners involved in” between persons and projects, however the relation is “involved in” between “partners” and “projects”.

Sometimes queries fail because of a combination of different reasons. For instance, “which main areas are corporately funded?”, falls within the category of ranking failure, but it is also a linguistic and conceptual failure (the latter because the ontology lacks a funding relationship between research-areas and corporations).

In sum, our estimation is that the implementation of the ranking services plus extending the NLP component to cover the basic queries not yet handled linguistically will address 14 of the 39 failures (35.89%). An implementation of new mechanisms to handle combinations of basic queries will address another 12 failures (30.7%). Hence, removing redundancies and including also the fixes to the ontology, with both implementations it will be possible to handle 34 of the 39 failures, thus potentially achieving a 87% hit rate for AquaLog. Although a more detailed analysis of these problems may be needed, at this stage it does not seem particularly problematic to add these functionalities to AquaLog.

5 Related Work

We already pointed out that research in natural language interfaces to databases is currently a bit ‘dormant’ (although see [26] for recent work in the area), therefore it is not surprising that most current work on question answering is somewhat different in nature from AquaLog. Natural language search engines such as AskJeeves [27] and

EasyAsk [28] exist, as well as systems such as START [13] and REXTOR [21], whose goal is to extract answers from text. AnswerBus [29] is another open-domain question-answering system based on web information retrieval. FAQ Finder [30] is a natural language question-answering system that uses files of FAQs as its knowledge base; it also uses WordNet to improve its ability to match questions to answers, using two metrics: statistical similarity and semantic similarity.

PRECISE [26] maps questions to the corresponding SQL query, by identifying classes of questions that are easy to understand in a well defined sense: the paper defines a formal notion of semantically tractable questions. Questions are sets of attribute/value pairs and a relation token corresponds to either an attribute token or a value token. Apart from the differences in terminology this is actually similar to the AquaLog model. In PRECISE, like in AquaLog, a lexicon is used to find synonyms. In PRECISE the problem of finding a mapping from the tokenization to the database is reduced to a graph matching problem. The main difference with AquaLog is that in PRECISE all tokens must be distinct, questions with unknown words are not semantically tractable and cannot be handled. In contrast with PRECISE, although AquaLog also uses pattern matching to identify at least one of the terms of the relation, it is still able in many cases to interpret the query, even if the words in the relation are not recognized (i.e., there is no match to any concept or relation in the ontology). The reason for this is that AquaLog is able to reason about the structure of the ontology to make sense of relations which appear to have no match to the KB. Using the example suggested in [26], AquaLog would not necessarily know the term “neighborhood”, but it might know that it must look for the value of a relation defined for cities. In many cases this information is all AquaLog needs to interpret the query.

MASQUE/SQL [7] is a portable natural language front-end to SQL databases. The semi-automatic configuration procedure uses a built-in domain editor which helps the user to describe the entity types to which the database refers, using an is-a hierarchy, and then declare the words expected to appear in the NL questions and define their meaning in terms of a logic predicate that is linked to a database table/view. In contrast with MASQUE/SQL AquaLog uses the ontology to describe the entities with no need for an intensive configuration procedure.

6 Conclusion

In this paper we have described the AquaLog query answering system, emphasizing its genesis in the context of semantic web research. Although only initial evaluation results are available, the approach used by AquaLog, which relies on a RSS component able to use information about the current ontology, string matching and similarity measures to interpret the intermediate queries generated by the NLP component, appears very promising. Moreover, in contrast with other systems AquaLog requires very little configuration effort. For the future we plan to make the AquaLog linguistic component more robust, primarily on the basis of the feedback received from the evaluation study carried out on the KMi domain. In addition we also intend to carry out a formal analysis of the RSS component to provide a more accurate and formal

account of its competence. As already mentioned, more comprehensive evaluation studies will also be needed. Finally, although the ontology-driven approach provides one of the main strength of AquaLog we have also started to investigate the possibility of accessing more than one ontology simultaneously in a transparent way for the user [31].

Acknowledgements. The authors would like to thank Maria Vargas-Vera and John Domingue for useful input on AquaLog and related topics. We are also grateful to Kalina Bontcheva for assistance with the use of the GATE NLP component. Finally we would like to thank all those members of the lab who took part in the preliminary evaluation of AquaLog.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American*, 284(5), 2001.
2. Dzbor, M., Domingue, J., Motta, E.: Magpie – Towards a Semantic Web Browser. Proceedings of the 2nd International Semantic Web Conference (ISWC2003), *Lecture Notes in Computer Science*, 2870/2003, Springer-Verlag, 2003
3. Mc Guinness, D.: Question Answering on the Semantic Web. *IEEE Intelligent Systems*, 19(1), 2004.
4. AKT Reference Ontology, <http://kmi.open.ac.uk/projects/akt/ref-onto/index.html>.
5. Burger, J., Cardie, C., Chaudhri, V., et al.: Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A). *NIST Technical Report*, 2001. PDF available from <http://www.ai.mit.edu/people/jimmylin/%0Apapers/Burger00-Roadmap.pdf>.
6. Kaplan, J.: Designing a portable natural language database query system. *ACM Transactions on Database Systems* 9(1), pp. 1-19, 1984.
7. Androultsopoulos, I., Ritchie, G.D., and Thanisch, P.: MASQUE/SQL - An Efficient and Portable Natural Language Query Interface for Relational Databases. In Chung, P.W. Lovegrove, G. and Ali, M. (Eds.), *Proceedings of the 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Edinburgh, U.K., pp. 327-330. Gordon and Breach Publishers, 1993.
8. Chu-Carroll, J., Ferrucci, D., Prager, J., Welty, C.: Hybridization in Question Answering Systems. In Maybury, M. (editor), *New Directions in Question Answering*, AAAI Press, 2003.
9. Jung, H., Geunbae Lee, G.: Multilingual Question Answering with High Portability on Relational Databases. *IEICE transactions on information and systems*, E86-D(2), pp306-315, Feb 2003.
10. Androultsopoulos, I., Ritchie, G.D., Thanisch P.: Natural Language Interfaces to Databases - An Introduction. *Natural Language Engineering*, 1(1), pp. 29-81, Cambridge University Press, 1995.
11. Hunter, A.: Natural language database interfaces. *Knowledge Management*, May 2000.
12. Vargas-Vera, M., Motta, E., Domingue, J.: AQUA: An Ontology-Driven Question Answering System. In Maybury, M. (editor), *New Directions in Question Answering*, AAAI Press (2003).

13. Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J., Marton, G., McFarland A. J., Temelkuran, B.: Omnidbase: Uniform Access to Heterogeneous Data for Question Answering.. *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB)*, 2002.
14. RDF: <http://www.w3.org/RDF/>.
15. Mc Guinness, D., van Harmelen, F.: OWL Web Ontology Language Overview. W3C Recommendation 10 (2004) <http://www.w3.org/TR/owl-features/>.
16. Motta, E.: *Reusable Components for Knowledge Modelling*. IOS Press, Amsterdam, The Netherlands. (1999).
17. WebOnto project: <http://plainmoor.open.ac.uk/webonto>.
18. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics ACL'02*. Philadelphia, 2002.
19. Tablan, V., Maynard, D., Bontcheva, K.: *GATE --- A Concise User Guide*. University of Sheffield, UK. <http://gate.ac.uk/>.
20. Katz, B., Lin, J.: Selectively Using Relations to Improve Precision in Question Answering. *Proceedings of the EACL-2003. Workshop on Natural Language Processing for Question Answering*, 2003.
21. Katz, B., Lin, J.: REXTOR: A System for Generating Relations from Natural Language. Proceedings of the ACL 2000 Workshop of Natural Language Processing and Information Retrieval (NLP&IR), 2000.
22. Klein, D. and Manning, C. D.: Fast Exact Inference with a Factored Model for Natural Language Parsing. *Advances in Neural Information Processing Systems 15*, 2002.
23. Fellbaum, C. (editor), *WordNet, An Electronic Lexical Database*. Bradford Books, May 1998.
24. Cohen, W., W., Ravikumar, P., Fienberg, S., E.: A Comparison of String Distance Metrics for Name-Matching Tasks. In *IIWeb Workshop 2003*, PDF available from <http://www-2.cs.cmu.edu/~wcohen/postscript/ijcai-ws-2003.pdf>, 2003.
25. Motta, E., Domingue, J., Cabral, L., Gaspari, M.: IRS-II: A Framework and Infrastructure for Semantic Web Services, 2nd International Semantic Web Conference (ISWC2003), *Lecture Notes in Computer Science*, 2870/2003, Springer-Verlag, 2003.
26. Popescu, A., M., Etzioni, O., Kautz, H., A.: Towards a theory of natural language interfaces to databases. *Proceedings of the 2003 International Conference on Intelligent User Interfaces*, January 12-15, 2003, pp. 149-157, Miami, FL, USA.
27. AskJeeves: <http://www.ask.co.uk>.
28. EasyAsk: <http://www.easyask.com>.
29. Zheng, Z.: The AnswerBus Question Answering System. *Proc. of the Human Language Technology Conference (HLT 2002)*. San Diego, CA. March 24-27, 2002.
30. Burke, R., D., Hammond, K., J., Kulyukin, V.: Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder system. *Tech. Rep. TR-97-05, Department of Computer Science, University of Chicago*, 1997.
31. Waldinger, R., Appelt, D. E., et al.: Deductive Question Answering from Multiple Resources. In Maybury, M. (editor), *New Directions in Question Answering*, AAAI Press, 2003.

Schema-Based Natural Language Semantic Mapping

Niculae Stratica and Bipin C. Desai

Department of Computer Science, Concordia University,
1455 de Maisonneuve Blvd. West Montreal, H3G 1M8, Canada
nstratica@primus.ca
bcdesai@cs.concordia.ca

Abstract. This paper addresses the problem of mapping Natural Language to SQL queries. It assumes that the input is in English language and details a methodology to build a SQL query based on the input sentence, a dictionary and a set of production rules. The dictionary consists of semantic sets and index files. A semantic set is created for each table or attribute name and contains synonyms, hyponyms and hypernyms as retrieved by WordNet and complemented manually. The index files contain pointers to records in the database, ordered by value and by data type. The dictionary and the production rules form a context-free grammar for producing the SQL queries. The context ambiguities are addressed through the use of the derivationally related forms based on WordNet. Building the run time semantic sets of the input tokens helps solving the ambiguities related to the database schema. The proposed method introduces two functional entities: a pre-processor and a runtime engine. The pre-processor reads the database schema and uses WordNet to create the semantic sets and the set of production rules. It also reads the database records and creates the index files. The run time engine matches the input tokens to the dictionary and uses the rules to create the corresponding SQL query.

1 Introduction

In early research work Minker [3] identified three basic steps in developing a natural language processing system. First, a model for the input language must be selected. Second, the input language must be analyzed and converted into an internal representation based on the syntactic analysis. Third, the internal representation is translated in the target language by using the semantic analysis. The syntactic and semantic analyzes can use learning algorithms and statistical methods [5] as shown in Fig. 1. The methods involved in the semantic analysis, either deterministic or statistical are heavily dependent on the context. There is always a degree of uncertainty related to the context and to the ambiguities of the natural language that translates into errors in the target language. Stuart, Chen and Shyu showed that the influence of the context on meaning grows exponentially with the length of a word sequence and it can be addressed through the so-called rule-based randomization [4]. Another way to minimize the uncertainty during the semantic analysis is to use highly organized data.

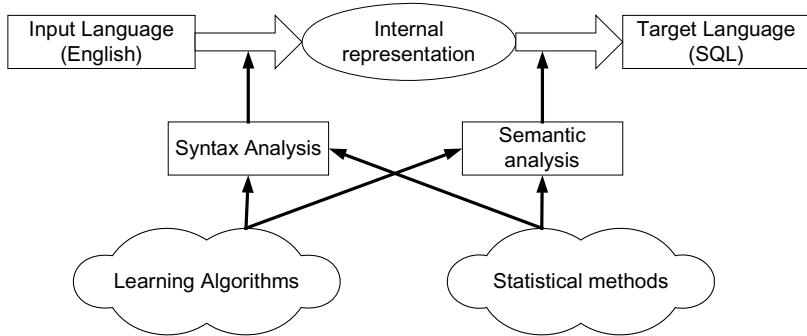


Fig. 1. Natural Language Processing through learning algorithms and statistical methods

The present paper describes a method for the semantic analysis of the natural language query and its translation in SQL for a relational database. It is a continuation of the work presented at NLDB'2003 conference in Cottbus, Germany [2].

2 Natural Language Processing and RDBMS¹

The highly organized data in RDBMS can be used to improve the quality of the semantic analysis. Previous work [2] introduced a template-based semantic analysis approach as shown on the left hand side of Fig. 2. The system consists of a syntactic analyzer based on the LinkParser [10] and a semantic analyzer using templates. The pre-processor builds the domain-specific interpretation rules based on the database schema, WordNet and a set of production rules. This method has limited capabilities in addressing the context ambiguities. The current goal is to improve the context disambiguation.

The method presented in this paper is shown on the right hand side of Fig. 2. The intent is to build the semantic analysis through the use of the database properties. Unlike the work described in [2], this method does not use syntactic analysis and it does not use user supplied rules. The input sentence is tokenized and then it is sent to the semantic analyzer. The semantic analysis is based on token matching between the input and a dictionary. The dictionary is formed of semantic sets and index files. The semantic sets are based on the synonyms, hypernyms and hyponyms related to the table and attribute names, as returned by WordNet [1]. The semantic set is manually edited to eliminate the less relevant elements and to address the case of meaningless names and abbreviations, such as Empl for Employers or Stdnt for Students. The index files are links to the actual records in the database. The production rules are based on the database schema and are used to generate the target SQL query as shown in Fig. 3.

¹ RDBMS Relational Database Management System

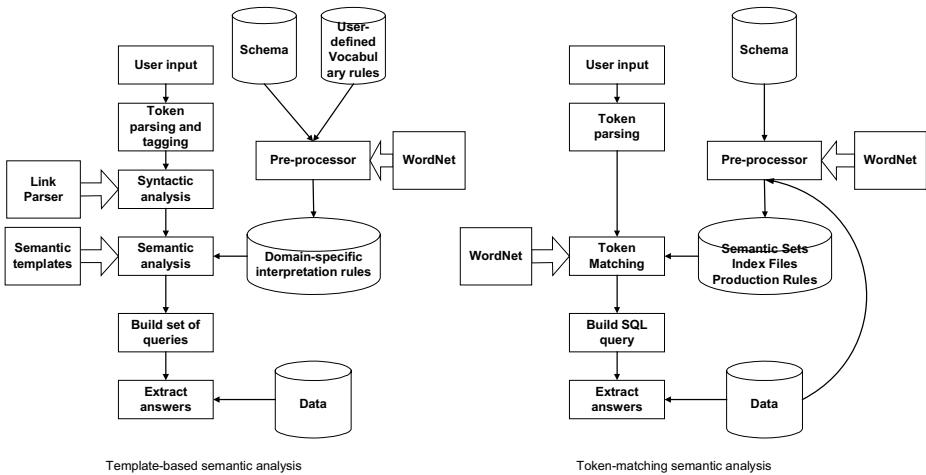


Fig. 2. Differences between the template-based and the token matching approaches

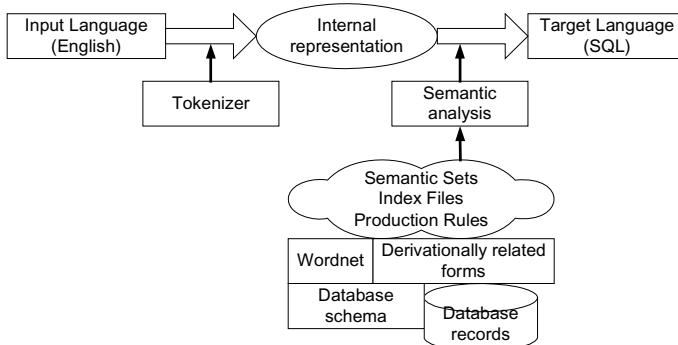


Fig. 3. Input language to SQL query

The internal representation is reduced to a set of tokens extracted from the natural language query. Using the derivationally related forms based on WordNet solves the run time ambiguities as shown in the following paragraphs.

3 The High Level Architecture

The proposed method is based on two functional units: a pre-processor and a runtime engine. The pre-processor analyzes the database schema and creates the semantic sets, the index files and the production rules, as shown in Fig. 4. The run time engine uses the semantic sets and the index files to match the input tokens with table and attribute names or with values in the database.

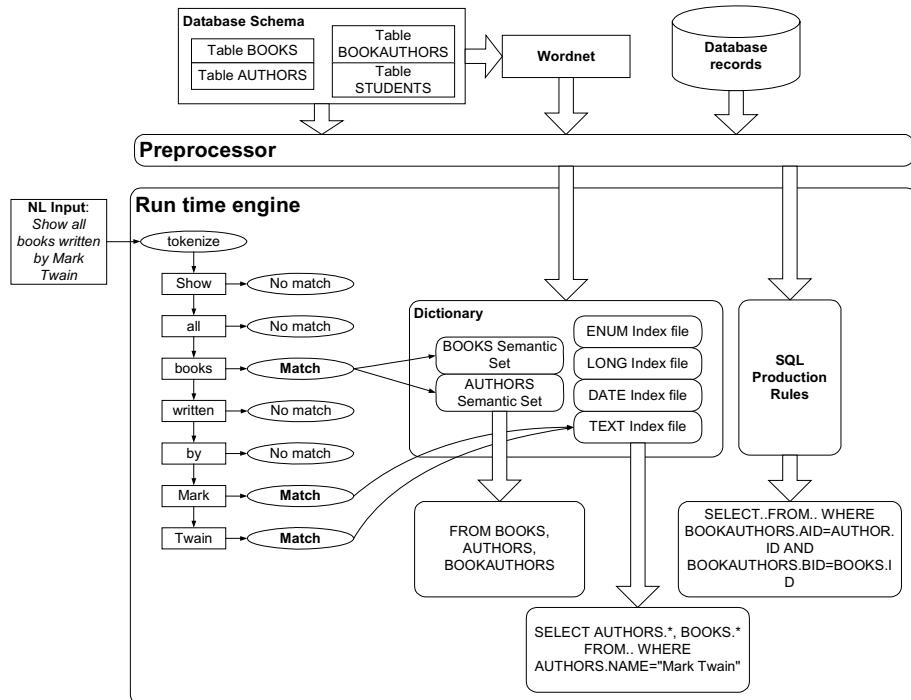


Fig. 4. The high level design

3.1 The Database Schema

In the example shown in Fig. 4. the database contains four tables as described below:

```

Table AUTHORS (LONG id PRIMARY_KEY, CHAR name, DATE DOB)
Table BOOKS (LONG id PRIMARY_KEY, CHAR title, LONG isbn, LONG pages,
DATE date_published, ENUM type)
Table BOOKAUTHORS (LONG aid references AUTHORS, LONG bid references
BOOKS)
Table STUDENTS (LONG id PRIMARY_KEY, CHAR name, DATE DOB)

Table AUTHORS (id, name, DOB)
1, 'Mark Twain', NOV-20-1835
2, 'William Shakespeare', APR-23-1564
3, 'Emma Lazarus', MAR-3-1849

Table BOOKS (id, title, isbn, pages, date_published, type)
1, 'Poems', 12223335, 500, FEB-14, SEP-1-1865, 'POEM'
2, 'Tom Sawyer', 12223333, 700, NOV-1-1870, 'ROMAN'
3, 'Romeo and Juliet', 12223334, 600, AUG-5-1595, 'TRAGEDY'

```

```
Table BOOKAUTHORS (aid, bid)
1, 2
2, 3
3, 1

Table STUDENTS (id, name, DOB)
1, 'John Markus', FEB-2-1985
2, 'Mark Twain', APR-2-1980
3, 'Eli Moss', SEP-10-1982
```

3.2 Using WordNet to Create the Semantic Sets

The database schema table and attribute names are introduced as input to WordNet² to build the semantic sets for each table and attribute name. The semantic sets include synonyms, hyponyms and hyponyms of each name in the schema. For table AUTHORS WordNet returns the following list:

```
WordNet 2.0 Search
Top of Form
Bottom of Form
Overview for 'author'
The noun 'author' has 2 senses in WordNet.
1. writer, author -- (writes (books or stories or articles or the
like) professionally (for pay))
2. generator, source, author -- (someone who originates or causes or
initiates something; 'he was the generator of several complaints')
Results for 'Synonyms, hypernyms and hyponyms ordered by estimated
frequency' search of noun 'authors'
2 senses of author
Sense 1 writer, author => communicator
Sense 2 generator, source, author => maker, shaper
```

The semantic set for AUTHORS becomes:

```
{writer, author, generator, source, maker, communicator, shaper}
```

Any token match between the input sentence and the AUTHORS semantic set elements will place the AUTHORS in the SQL table list. WordNet returns the following semantic set for BOOKS:

```
{book, volume, ledger, leger, account book, book of account, record,
record book, script, playscript, rule book}
```

Similarly all the remaining table and attribute names in the database schema are associated with their semantic sets as returned by WordNet. One limitation of the model resides in the necessity of having meaningful names for all attributes and tables. If this is not the case, the corresponding semantic sets have to be built manually, by sending the appropriate queries to WordNet. For example, instead of using the AUTHORS attribute name 'DOB' the operator queries WordNet for 'Date' and adds 'birth date' to the semantic set associated with AUTHORS.DOB.

² WordNet Reference: <http://www.cogsci.princeton.edu/~wn/> [1]

3.3 Indexing the ENUM Types

An index file relating (Table, Attribute) and the ENUM values is created for each ENUM attribute. Example: TABLE=BOOKS, ATTRIBUTE=TYPE, ENUM={ROMAN, NOVEL, POEM} If any of the TYPE values occurs in the input sentence, a new production rule is added to the SQL relating BOOKS.TYPE to VALUE such as the one in the example below:

```
Sentence: 'List all romans'
ROMAN is a valid value for BOOK.TYPE
The production rule is:
WHERE BOOKS.TYPE='ROMAN'
The resulting SQL query is:
SELECT BOOKS.* FROM BOOKS WHERE BOOKS.TYPE='ROMAN'
```

3.4 Indexing the TEXT Type

A compressed BLOB³ relating (Table, Attribute) and the attribute values is created for each TEXT attribute. Example: TABLE=AUTHORS, ATTRIBUTE=NAME, TEXT={'Mark Twain', 'William Shakespeare', 'Emma Lazarus'} If any of the indexed NAME values occurs in the input sentence, a new production rule is added to the SQL relating AUTHORS.NAME to VALUE as in the example below:

```
Sentence: 'Show information about Mark Twain'
'Mark', 'Twain' are values related to AUTHORS.NAME
The production rule is:
AUTHORS.NAME='Mark Twain'
The resulting SQL query is:
SELECT AUTHORS.* FROM AUTHORS WHERE AUTHORS.NAME='Mark Twain'
```

3.5 Building the SQL Production Rules

The database schema is used to build a set of SQL query elements involving relations to be joined. For example the tables AUTHORS and BOOKS participate in a N-to-N relationship with the table BOOKAUTHORS as shown in the database schema given in section 3.1.

The pre-processor builds the following production rule:

```
IF AUTHORS in Table List AND BOOKS in Table List Then BOOKAUTHORS is
in Table List
```

and the following SQL template:

```
SELECT Attribute List FROM Table List
WHERE BOOKAUTHORS.AID=AUTHORS.ID AND BOOKAUTHORS.BID=BOOKS.ID
```

³ BLOB Binary Large Object

3.6 The Run Time Workflow

Let us consider Fig. 4., where the input sentence is ‘Show all books written by Mark Twain’. Based on the dictionary and on the production rules, the run time engine incrementally builds the attribute list, the table list and the SQL constraints as shown above. The tokens from the input sentence are matched with elements in the semantic sets or in the index files as show bellow:

```
'Show' = no match
'all' = no match
'books' = matches one entry in the semantic set BOOKS
'written' = no match
'by' = no match
'Mark' = matches one entry in the index file associated with
AUTHORS.NAME
'Twain' = matches one entry in the index file associated with
AUTHORS.NAME
```

The token ‘books’ is matched with table BOOKS. Because ‘Mark’ and Twain’ are neighbors in the input query and because they both point to the same value of the AUTHORS.NAME attribute in that order, they are merged into ‘Mark Twain’ and the correspondig SQL constraint becomes AUTHORS.NAME=‘Mark Twain’. As shown in section 3.5 the tables AUTHORS and BOOKS form a N-to-N relation along with table BOOKAUTHORS. The three tables are included in the Table List. Because tables BOOK and AUTHOR are referenced in the matches, they are both included in the Attributes List:

BOOKS, AUTHORS, BOOKAUTHORS (1)

and the first two SQL constraints are:

BOOKAUTHORS.AID=AUTHORS.ID AND BOOKAUTHORS.BID=BOOKS.ID (2)

An additional SQL constraint is:

AUTHORS.NAME=‘Mark Twain’ (3)

Based on (1), (2) and (3) the resulting SQL query is:

```
SELECT AUTHORS.*, BOOKS.* FROM AUTHORS, BOOKS, BOOKAUTHORS
WHERE BOOKAUTHORS.AID=AUTHORS.ID AND BOOKAUTHORS.BID=BOOKS.ID
AND AUTHORS.NAME=‘Mark Twain’
```

In the model presented above all attributes of the primary tables AUTHORS and BOOKS are in the SQL attribute list and the attributes of the table BOOKAUTHORS are not.

3.7 The Ambiguity Resolution for Twin Attributes Values

If two tables share the same attribute value as in AUTHORS.NAME=‘Mark Twain’ and STUDENTS.NAME=‘Mark Twain’ then there is ambiguity to whether BOOKS should be related to AUTHORS or to STUDENTS. The ambiguity is solved at run time. The token ‘written’ from the input sentence is processed through WordNet for Coordinate Terms. Here are the results:

```
Results for 'Derivationally related forms' search of verb 'written'
9 senses of write
Sense 1
    RELATED TO->(noun) writer#1
        => writer, author -
    RELATED TO->(noun) writing#1
        => writing, authorship, composition, penning
```

The dynamic semantic set associates ‘written’ to {writer, author, writing, authorship, composition, penning}. Comparing the elements of this set to the preprocessed semantic sets, it results that ‘written’ is related to AUTHORS set and not to the STUDENTS.

3.8 The Context Ambiguity Resolution

Early work [8] showed how a number of different knowledge sources are necessary to perform automatic disambiguation. Others [7] used restrained input language and syntactic constructions that were allowed. Yet another approach is to introduce stochastic context definition as presented in [9]. The present paper uses the database schema and WordNet to address the context ambiguity.

Let us consider the tables described in Section 3.1 and the new table introduced below:

```
Table BORROWEDBOOKS (LONG bid references BOOKS, LONG sid references
STUDENTS)
```

Let the input sentence be ‘Who borrowed Romeo and Juliet?’ Following the workflow shown in section 3.6, the analyzer returns only this match:

```
BOOKS.NAME=‘Romeo and Juliet’
```

The table list includes only BOOKS and the resulting SQL query is:

```
SELECT BOOKS.* FROM BOOKS
WHERE BOOKS.NAME=‘Romeo and Juliet’
```

The returned result set shows all attribute values for the selected book, however it does not show who borrowed the book, if this is the case. There are two approaches

for fixing the ambiguity: a. all table and attribute names in the database schema have meaningful names and b. the table BORROWEDBOOKS receives the semantic set of the token BORROWED. Both option require manual operations before or during the pre-processing time. The run time results are:

```
'borrowed' is matched with the semantic set for BORROWEDBOOKS
BORROWEDBOOKS table involves BOOKS and STUDENTS tables
BOOKS and STUDENTS are in N-to-N relationship represented by the table BORROWEDBOOKS
The Table List becomes BOOKS, STUDENTS, BORROWEDBOOKS
From the database schema the first SQL constraint is BORROWED-
BOOKS.BID=BOOKS.ID AND BORROWEDBOOKS.SID=STUDENTS.ID
From the TEXT index matching the second SQL constraint is
BOOKS.NAME='Romeo and Juliet'
The attribute list is BOOKS.*, STUDENTS.*
The final SQL query is:
SELECT BOOKS.*, STUDENTS.* FROM BOOKS, STUDENTS, BORROWEDBOOKS WHERE
BORROWEDBOOKS.BID=BOOKS.ID AND BORROWEDBOOKS.SID=STUDENTS.ID AND
BOOKS.NAME='Romeo and Juliet'
```

The result set shows the required information, if any.

4 Capabilities and Limitations

The proposed method accepts complex queries. It can disambiguate the context if there are enough elements in the input that are successfully matched, as in the example shown bellow:

```
'Show all romans written by Mark Twain and William Shakespeare that
have been borrowed by John Markus'
```

The method retains the following tokens:

```
'... ... romans ... ... Mark Twain ... William Shakespeare ... ... ...
borrowed ... John Markus'
```

The token 'romans' point to table BOOKS. 'romans' is found in the INDEX files for ENUM values of the attribute BOOKS.TYPE. Following the workflows presented in 3.6 and 3.8, the method findings are:

```
'romans' matches the ENUM value BOOKS.TYPE='ROMAN'
'Mark Twain' matches the AUTHORS.NAME='Mark Twain'
'William Shakespeare' matches the AUTHOR.NAME='William Shakespeare'
'borrowed' is disambiguated at run time through BORROWEDBOOKS
'John Markus' matches STUDENTS.NAME='John Markus'
Schema correlates AUTHORS, BOOKS and BOOKAUTHORS
Schema correlates STUDENT, BOOKS and BORROWEDBOOKS
The table list becomes: AUTHORS, BOOKS, BOOKAUTHORS, STUDENTS,
BORROWEDBOOKS
```

The SQL Constraints are:

```
BOOKAUTHORS.AID=AUTHORS.ID AND BOOKAUTHORS.BID=BOOKS.ID
AND BOOKS.TYPE='ROMAN'
AND (AUTHORS.NAME='Mark Twain' OR AUTHORS.NAME='William Shakespeare')
AND STUDENTS.NAME='John Markus'
AND BOOKS.ID=BORROWEDBOOKS.BID AND STUDENT.ID=BORROWEDBOOKS.SID
```

The two constraints to the AUTHORS.NAME have been OR-ed because they point to the same attribute. The method allows to construct the correct SQL query. As has already been shown in 3.7 and 3.8, the method can address context and value ambiguities.

The current architecture, however, does not support operators such as: greater than, less than, count, average and sum. It does not resolve dates as in: before, after, between. The generated SQL does not support imbricated queries. The proposed method eliminates all tokens that cannot be matched with either the semantic sets or with the index files and it works for semantically stable databases. The prerpocessor must be used after each semantic update of the database in order to modify the index files. The context disambiguation is limited to the semantic sets related to a given schema. Errors related to tokenizing, WordNet and the human intervention propagate in the SQL query. The method completely disregards the unmatched tokes and thus it cannot correct the input query if it has errors. However, the method correctly interprets the tokens that are found in the semantic sets or among the derivationally related terms at run time.

5 Future Work

The future work will focus on the operator resolution as listed in the section 4. We believe that the approach presented in this paper can give good results with a minimum of effort in implementation and avoids specific problems related to the various existing semantic analyses approaches. This is partly made possible by the highly organized data in the RDBMS. The method will be implemented and the results will be measured against complex sentences involving more than 4 tables from the database. A study will be done to show the performance dependency on the size of the database records and on the database schema.

Acknowledgements. The present work was made possible thanks to Dr. Leila Kosseim at Concordia university, who gave advice and supervision throughout the development of the method and participated in the previous work related to Natural Language interface for database [2] to which this paper is related.

References

1. Miller, G., WordNet: A Lexical Database for English, Communications of the ACM, 38 (1), pp. 39-41, November 1995
2. N. Stratica, L. Kosseim and B.C. Desai, NLIDB Templates for Semantic Parsing, Proceedings of Applications of Natural Language to Data Bases, NLDB'2003, pp. 235-241, June 2003, Burg, Germany.
3. Minker, J., Information storage and retrieval - a survey and functional description, SIGIR, 12, pp.1-108, 1997
4. Stuart H. Rubin, Shu-Ching Chen, and Mei-Ling Shyu, Field-Effect Natural Language Semantic Mapping, Proceedings of the 2003 IEEE International Conference on Systems, Man & Cybernetics, pp. 2483-2487, October 5-8, 2003, Washington, D.C., USA.
5. Lawrence J. Mazlack, Richard A. Feinauer, Establishing a Basis for Mapping Natural-Language Statements Onto a Database Query Language, SIGIR 1980: 192-202
6. Allen, James, Natural Language Understanding, University of Rochester 1995 The Benjamin Cummings Publishing Company, Inc. ISBN: 0-8053-0334-0
7. Kathryn Baker, Alexander Franz, and Pamela Jordan, *Coping with Ambiguity in Knowledge-based Natural Language Analysis*, Florida AI Research Symposium, pages 155-159, 1994 Pensacola, Florida
8. Hirst G., Semantic Interpretation and the Resolution of Ambiguity, Cambridge University Press 1986, Cambridge
9. Latent Semantic Analysis Laboratory at the Colorado University, <http://lsa.colorado.edu/> site visited in March 2004
10. Sleator D., Davy Temperley, D., Parsing English with A Link Grammar,Proceedings of the Third Annual Workshop on Parsing Technologies, 1993

Avaya Interactive Dashboard (AID): An Interactive Tool for Mining the Avaya Problem Ticket Database

Ziyang Wang¹ and Amit Bagga²

¹ Department of Computer Science

New York University

New York, NY 10012

ziyang@cs.nyu.edu

² Avaya Labs Research

233 Mt Airy Road Basking Ridge, NJ 07920

bagga@avaya.com

Abstract. In this paper we describe an interactive tool called the Avaya Interactive Dashboard, or AID, that was designed to help Avaya’s services organization to mine the text fields in Maestro. AID allows engineers to quickly and conveniently drill down to discover patterns and/or verify intuitions with a few simple clicks of the mouse. The interface has a web-based front-end that interacts through CGI scripts with a central server implemented mostly in Java. The central server in turn interacts with Maestro as needed.

1 Introduction

The Avaya problem database (Maestro) consists of approximately 5 million records. Each record in the database corresponds to a problem in one of Avaya’s deployed products and is populated either by the product itself (called alarms) via a self-diagnostic reporting mechanism, or by a service engineer via a problem reporting phone call. In the latter case, a human operator listens to the problem described by the engineer and creates the database record (called tickets) manually. Presently, the Maestro database consists of approximately 4 million alarms and a million tickets.

Each record in Maestro consists of several structured fields (name of customer, location, date of problem, type of product, etc.), and at least three unstructured text fields: problem description, one or more notes on the progress, and problem resolution description. The unstructured fields are restricted to 256 bytes and this constraint forces the operators summarize and abbreviate liberally. As the problem is worked on by one or more engineers, the notes fields are updated accordingly. Upon resolution, the resolution field is updated. All updates to the text fields occur via the phone operators. The heavy use of summarization and abbreviations results in the data contained in these fields to be “dirty.” In other words, there are numerous typos, uses of inconsistent abbreviations, non-standard acronyms, etc. in these fields.

The database is mined regularly by Avaya's researchers and services engineers for discovering various patterns such as: frequency of alarms and tickets, time of occurrence, number of alarms or tickets per product or customer, etc. Thus far, only the structured fields were mined. However, there was a great need for mining the unstructured text fields as they contain rich information like the type of problem, the solution, etc. In fact, the unstructured data fields were being mined manually by a set of service engineers by first using database queries on the structured fields (to restrict the number of tickets returned) and then by manually eye balling the resulting tickets to discover patterns like types of problems, commonly occurring problems by customer/location, and commonly occurring problems across customers/locations for a particular product, etc.

In this paper we describe an interactive tool called the Avaya Interactive Dashboard, or AID, that was designed to help Avaya's services organization to mine the text fields in Maestro. AID allows engineers to quickly and conveniently drill down to discover patterns and/or verify intuitions with a few simple clicks of the mouse. The interface has a web-based front-end that interacts through CGI scripts with a central server implemented mostly in Java. The central server in turn interacts with Maestro as needed.

2 The Design of AID

AID provides search utilities on large-scale relational database with focus on text analysis and mining. As mentioned earlier, through automatic text analysis, AID allows service engineers to quickly and conveniently discover patterns and verify intuitions about problems. The design of this application complies with several objectives. In this section, we first briefly discuss these designing objectives, then we intensively discuss issues related to them, including functionalities, algorithms and application model.

2.1 Design Goals

AID emphasizes on three designing issues to provide advanced service: usability, precision and performance. High usability is a fundamental requirement for service oriented applications. It has two underline meanings. First, the services must be easy to use. The application interface should be simple and well ported on highly available system such as the Internet. And second, it should provide an array of functions for supporting different tasks. In general, search and mining utilities can simplify the tasks of automatically identifying relevant information or providing more specific information. Our design of functionalities are based on these two tasks with several additional features. Precision is a standard evaluation metric for search utilities and text analysis. The algorithms we use in our approach must achieve high precision to help users directly target the problems. The TFIDF model in information retrieval is used for text similarity analysis and hierarchical clustering is used to group results respectively. In considering the performance requirement, we want our application to provide quick and highly

available services. AID is a centralized server application. The server must respond to queries quickly to achieve high availability if requests aggregate.

2.2 Functionalities

AID has two major functionalities: searching relevant tickets using keywords or sample tickets, and clustering a set of tickets into groups to identify more specific information. Keyword and sample ticket search is based on the text fields of tickets in the relational database. It can be constrained by non-text fields of tickets. Clustering is a standard method in data mining which may be well used on relational database. Our approach is very specific that it only performs clustering on text fields. Furthermore, two additional functions are provided: formatted retrieval of single ticket and categorizing tickets by non-text fields. These functionalities are well linked and one can be performed based the results of one another. For example, clustering can be performed on the results given by a keyword search and for each of the groups in clustered results, categorization can be applied. By combining a sequence of operations, such interactive functions give rich and flexible choices to identify problems very specific to users.

2.3 Algorithms

We use two well-studied algorithms in information retrieval and mining, TFIDF and hierarchical clustering, as part of the text analysis modules in AID.

TFIDF. This relevance algorithm is originally proposed by Rocchio [4] for the vector space retrieval model [5]. In the vector space model, a text document is viewed as a bag of terms (or words) and represented by a vector in the vocabulary space. The fundamental algorithm of TFIDF chooses to compute the relevance score of two text documents by summing the products of weights for each term that appears in both text documents, normalized by the product of Euclidean vector length of the two text documents. The weight of a term is a function of term occurrence frequency (called *term frequency (TF)*) in the document and the number of documents containing the term in collection (the *inverse document frequency (IDF)*). Mathematically, let N be the number of text documents in collection, x_{ij} be the indicator of whether term i occurs in document j ($x_{ij} = 1$ if term i in document j or 0 otherwise), the inverse document frequency of term i is defined as

$$IDF_i = \log\left(\frac{N}{\sum_{j=1}^N x_{ij}}\right). \quad (1)$$

And the relevance score of two text documents are given as

$$R_{ij} = \frac{\sum_k w_{ki} \times w_{kj}}{\sqrt{\sum_k w_{ki}^2} \times \sqrt{\sum_k w_{kj}^2}} \quad (2)$$

where $w_{kl} = TF_{kl} \times IDF_l$.

Hierarchical clustering based on text similarity. There are several different clustering algorithms which has been studied intensively [2]. Hierarchical clustering is a bottom-up approach based on the distance of each pair of elements in the beginning and distance of each pair of clusters in the intermediate process. Since clustering is also performed on text fields, we use *TFIDF* as the similarity measure to compute distance. Hierarchical clustering has performance in the order of square.

2.4 Application Model

AID is a centralized server application, which may be used by multiple users simultaneously. It is authorized to read data on relational database but not authorized to write to the database. The server maintains statelessness for multiple requests without caching any history operations performed by a certain user. If any task requires stateful transition, the client must provide complete information of the whole procedure for server to determine which action to perform. Such application model is very similar to a simple web server. Many implementation and optimization techniques in building web servers can be safely used here to enhance performance. The main difference from a web server is that the backend is a database, not a file system. The performance of the server may be limited by the capacity of database query processing and data transferring rate.

3 Implementation

AID is a 3-tier server application: an independent interface, a multifunctional server and a backend database. We developed a web interface for AID to facilitate the using of AID search service. A CGI program accepts requests from the web and re-formats them compatible with the interface of AID server. AID server is the central service provider which organizes the internal logic and perform proper analysis. The backend database is the read-only data source to AID. The main code of AID is written in Java with minor combination with C. Since our motivation is not to develop a search utility serving a very large population, Java performance can properly satisfy our needs.

3.1 The Architecture of AID Server

In the basic scenario, AID server accepts and processes incoming requests, retrieves data from the database, performs text analysis and sends out results in HTML format. Figure 1 shows the server components in the view of data flow. The server socket module maintains a pool of working threads.

Each incoming request is assigned a working thread and some necessary resources. The request then is forwarded to the query manager which identifies which functionality is requested and pre-processes the request parameters before database retrieval. Different functionalities require different parameters. In searching relevant tickets, the searching can be constrained by several non-text

fields, such as product name/ID, customer name/ID, date, severity level. In clustering a set of tickets, a number of ticket identifiers have to be provided. These parameters are corrected and formatted to be compatible with database formats. The database module is responsible for querying and retrieving data from database. Since we allow multi-request simultaneous processing, we have implemented a high level SQL query processor which can handle multiple queries in parallel. Internally, we only maintain a few connections to database but multiple JDBC statements to handle queries. This approach has several good features:

- Good utilization the network bandwidth between AID server and database. Since AID server is the only agent to communicate with database when multiple users are using AID service, a few connections are sufficient to transfer data over network.
- Avoids overflow of number of database connections. If each working thread processing a request occupies a connection to database, the database connections will be overburdened and the overhead for initializing many connections is significant.
- Queries can be executed simultaneously using multiple JDBC statements over a few connections. The database schedules data transferring based on availability. So in a given connection, the data for any query will not block even if the data of previous queries is not ready.

Text analysis module is the kernel component in our application. Many techniques of text processing are used here to provide high quality search service. We will discuss it in detail in the next subsection. When results are ready, the response module outputs HTML in a unified format in order to present results to users with a consistent look and feel.

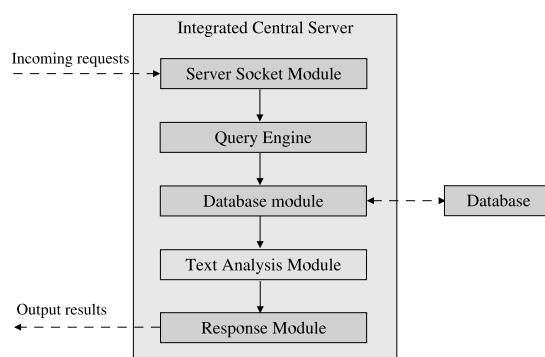


Fig. 1. AID server.

3.2 The Architecture of Text Analysis Module

Most of the functionalities provided by AID rely on text analysis. Figure 2 presents the architecture of text analysis module in AID server. All original text data, both text data retrieved from database and the given keywords, must be cleaned and translated by the text filter module. The module removes common stop words and performs a dictionary lookup for expanding the most commonly used abbreviations. of many abbreviations which are frequently used by technicians who write the text data. For example, “TOOS” is the abbreviation of “Totally Out Of Service.” When computing relevant tickets, the keywords/sample ticket and the retrieved tickets are passed into Relevance Evaluator to compute relevance score. However, for clustering or categorizing, the filtered data goes directly into Clustering module or categorizing module. Both Relevance Evaluator and Clustering make use of the TFIDF module which computes the similarity of two pieces of texts. Relevance searching requires 1-to-N similarity computation while hierarchical clustering is N-to-N computation. Therefore we present two different interfaces in TFIDF module accordingly. Once the Relevance Evaluator computes a score, it forwards the result immediately to Output Manager. The Output Manager module maintains an array of tickets with limited size. Only those with top relevance scores are put in the array. In this way, we cache only a small portion of retrieved data for memory efficiency. Finally the results are sorted for output.

It is worthwhile discussing the effect of document frequency in computing relevance score in TFIDF algorithm. Online computation of document frequencies is very expensive because it requires the complete scan of the database. Alternatively, we offline scan the database to collect them and regularly update them while AID server is running. The vocabulary in the collection contains 10,708 different terms which are cached in memory. As lookup is the only operation

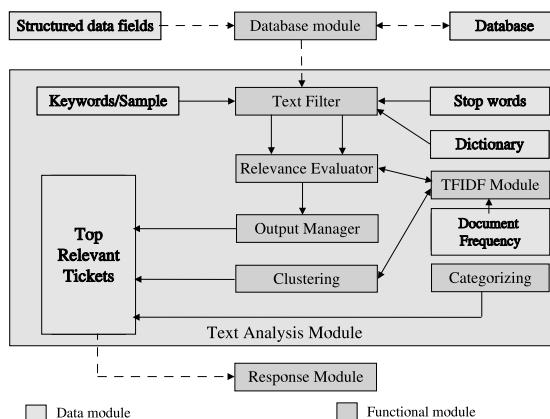


Fig. 2. The Text Analysis Module in AID server.

on document frequencies, the data is stored in a hash table for optimal lookup efficiency. AID can be configured to search in a subset of the database.

3.3 Multi-thread Model for Server Development

Our read-only server development is similar to a single web server where the backend data source comes from a database. Pai [3] discusses the properties of several architectures in building a portal web server and shows the overhead of multi-thread model mostly comes from the disk synchronization in file system and scheduling of threads in operating system. AID server has little use of local file system after startup so that it does not introduce much overhead like multi-thread web servers. Furthermore, we use a thread pool maintaining working threads so that when requests aggregate AID server does not initiate many new threads and the overhead for scheduling threads is greatly reduced.

4 Example

The usefulness of AID is best demonstrated with a detailed example. Figure 3 shows how AID can be useful in helping mine the text fields of Maestro. The first box in the figure shows the sample ticket that is used to search the database (for similar tickets). The sample ticket describes a platinum customer whose paging system is totally out of service. There is no overhead music. The ticket

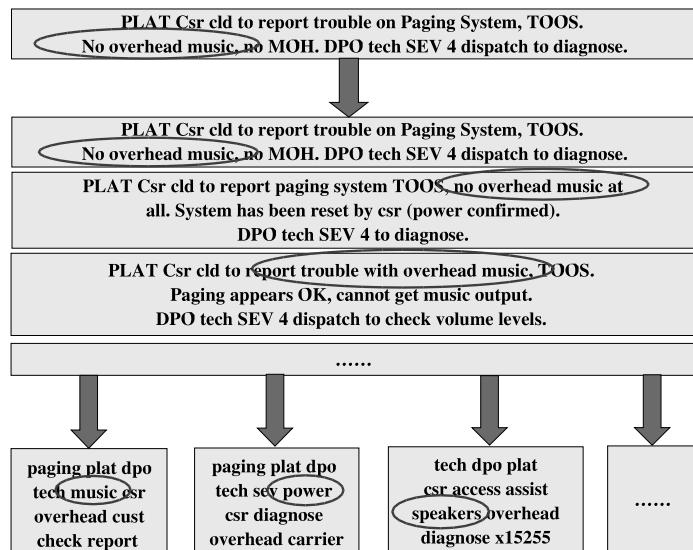


Fig. 3. Example on the use of AID

is assigned a severity value of 4 (the highest possible) and an engineer has been dispatched to look at the problem. The second box shows the top 3 tickets that were returned using AID's search capabilities. All these tickets show a platinum customer having trouble with their paging system (totally out of service with no overhead music).

Using AID's clustering algorithms on the set of tickets returned as a result of the search shows a trend in the types of problems occurring in the paging systems. There are three main types of paging system problems: music, power, and speakers (box 3).

A manual analysis of the tickets returned by the search routines show that a large number of tickets report problems with a platinum customer (indicating a top customer). AID allows the user to categorize the tickets by customer, location, product, etc. Using the categorization function of AID by customer shows that the platinum customers referred to in all the tickets in box 2 are the same. If the user wants to dig deeper and analyze what locations are affected he/she can re-categorize the results in box 2 by location. Moreover, the user has the option of using the categorization feature to verify if there is a recurrence of a particular problem type (as shown in box 3) at a particular location.

Suppose, armed with the additional knowledge that platinum customer ABC Corporation is experiencing problems with their paging systems, the user wants to investigate further. The investigation now can take several possible routes:

1. The user may want to narrow the search to only ABC Corporation and search for all paging system problems there. This is simply done by returning to box 1 and by placing a restriction on the customer (in addition to providing the sample ticket).
2. The user may want to use assess the timelines of the problems, both at ABC Corporation and also for paging systems in general. This is done by returning to box 1 and placing a restriction on date ranges, product types, and/or customers.
3. The user may want to discover if, for any of these cases, there were any prior indication that a complete outage (severity 4) was going to occur. The indications may have been prior tickets with lower severity codes (such as music being intermittent, etc.) that were called in. Once again, the user can return to box 1 and search for tickets by placing a restriction on the severity code.

5 Evaluation

Given the fact that Maestro contains approximately one million tickets, a comprehensive evaluation of the interface is not possible. However, we have solicited comments from a few users of the interface and are in the process of incorporating changes based upon their feedback. The main benefit of using AID has been the increased productivity as the tool helps the user to quickly and conveniently drill down to the desired level. The slowest part of the system, the clustering module, takes under a minute to cluster a few hundred tickets.

6 Conclusion

In this paper we have described an interactive tool called Avaya Interactive Dashboard, or AID, that allows for quick and convenient mining of Avaya's problem ticket database. The web-based front end of the system interacts with the user and formats the users' queries for the central server which in turn interacts with the problem ticket database. The interface has helped improve the quality and the productivity of mining the unstructured text fields greatly.

References

1. V. Hristidis and Y. Papakonstantinou. Discover: keyword search in relational database. In *Proc. 28th VLDB Conference*, 2002.
2. A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3), September 1999.
3. V. S. Pai, P. Druschel, and W. Zwaenepoel. Flash: an efficient and portable web server. In *Proc. of the 1999 USENIX Annual Technical Conference*, 1999.
4. J. Rocchio. Relevance feedback in information retrieval. In *in Salton: The SMART Retrieval System: Experiments in Automatic Document Processing, Chapter 14, pages 313-323, Prentice-Hall*, 1971.
5. G. Salton. Developments in automatic text retrieval. *Science*, 253:974–979, 1991.

Information Modeling: The Process and the Required Competencies of Its Participants

P.J.M. Frederiks¹ and T.P. van der Weide²

¹ IT-N, Philips Semiconductors Nijmegen, The Netherlands
paul.frederiks@philips.com

² Nijmegen Institute for Computing and Information Science,
Radboud University Nijmegen, The Netherlands
tvdw@cs.kun.nl

Abstract. In recent literature it is commonly agreed that the first phase of the software development process is still an area of concern. Furthermore, while software technology has been changed and improved rapidly, the way of working and managing this process have remained behind. In this paper focus is on the process of information modeling, its quality and the required competencies of its participants (domain experts and system analysts). The competencies are discussed and motivated assuming natural language is the main communication vehicle between domain expert and system analyst. As a result, these competencies provide the clue for the effectiveness of the process of information modeling.

1 Introduction

Nowadays many methods exist for the development process of software. A number of examples are: *Iterative development* ([1], [2]), *Evolutionary development*, *Incremental development* ([3]), *Reuse-oriented development* ([4]) and *Formal systems development* ([5]).

As different as all these development processes may be, there are fundamental activities common to all. One of these activities is *requirements engineering* (RE), although this activity has its own rules in each development method. RE is the process of discovering the purpose for which the software system is meant, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, negotiation, decision-making (see [6]) and subsequent implementation. For an extensive overview of the field of RE we refer to [7].

Experts in the area of software engineering do agree that RE is the most important factor for the success of the ultimate solution for reasons that this phase closes the gap between the concrete and abstract way of viewing at phenomena in application domains ([8], [9]). As a consequence, during the RE process, the involved information objects from the Universe of Discourse (UoD) have to be identified and described formally. We will refer to this process as *Information Modeling (IM)*. The resulting model will serve as the common base for understanding and communication, while engineering the requirements.

Where different areas of expertise meet, natural language may be seen as the base mechanism for communication. It is for this reason that each general modeling technique should support this basis for communication to some extent. As a consequence, the quality of the modeling process is bounded by the quality of *concretizing into an informal description* augmented with the quality of *abstracting from this description*.

In this paper focus is on information modeling as an exchange process between a domain expert and a system analyst. Our intention is to describe this exchange process, and the underlying assumptions on its participants. Using natural language in a formalized way can, for example, be seen as a supplement to the concept of *use cases* (see [10]).

Roughly speaking, a domain expert can be characterized as someone with (1) superior detail-knowledge of the UoD but often (2) minor powers of abstraction from that same UoD. The characterization of a system analyst is the direct opposite. We will describe the required skills of both system analysts and domain experts from this strict dichotomy and pay attention to the areas where they (should) meet. Of course, in practice this separation is less strict. Note that as a result of the interaction during the modeling process the participants will learn from each other. The system analyst will become more or less a domain expert, while the domain expert will develop a more abstract view on the UoD in terms of the concepts of the modeling technique. This learning process has a positive influence on effectiveness and efficiency of the modeling process, both qualitative (in terms of the result) and quantitative (in terms of completion time).

According to [11] the IM process is refined into four more detailed phases. Note that this process is not necessarily to be completed before other phases of system development, i.e. design and implementation, can start. At each moment during this process the formal specification may be realized. Still the IM process may be recognized in the various methods for system development as discussed in the beginning of this section.

In order to initiate the information modeling process the *system analyst* must elicit an initial problem specification from *domain experts*. This is referred to as *requirements elicitation*. In this phase domain knowledge and user requirements are gathered in interactive sessions with domain experts and system analysts. Besides traditional techniques, more enhanced techniques may be applied, e.g. *cognitive* or *contextual* techniques. For more elicitation techniques, see [7] or [12].

The requirements elicitation results in an *informal specification*, also referred to as the *requirements document*. As natural language is human's essential vehicle to convey ideas, this requirements document is written in natural language. In case of an evolutionary development, the previous requirements document will be used as a starting point.

In an iterative process of *modeling*, *verification* and *validation* the informal specification evolves to a complete *formal specification*, also referred to as a *conceptual model*. The primary task of the system analyst is to map the sentences of this informal specification onto concepts of the particular conceptual *modeling technique* used. As a side product, a sample population of the concepts derived

from the example instances may be obtained. Using the formal syntax rules of the underlying modeling technique, the formal specification can be verified. The conceptual model in turn can be translated into a comprehensible format. For some purposes a prototype is the preferable format, for other purposes a description is better suited. In this paper we restrict ourselves to a description in terms of natural language sentences that is to be validated by the domain expert. The example population serves as a source when instantiated sentences are to be constructed, thereby creating a feedback loop in the IM process. This translation process is called *paraphrasing*.

Basically, the conceptual model may be seen as a generative device (grammar) capable to generate not only the informal specification, but also all other feasible states of the UoD.

The *correctness* of this way of working depends on whether the formal specification is a proper derivate of the informal specification which in its turn must be a true reflection of the UoD. Being a proper derivate is also referred to as the *falsification principle*, which states that the derived formal specification is deemed to be correct as long as it does not conflict with the informal specification. Being a true reflection is referred to as the *completeness principle*. It is falsified when a possible state of the UoD is not captured, or when the grammar can derive an unintended description of a UoD state.

These two principles require the participants of the modeling process to have some specific competencies. For instance, a domain expert should be able to come up with significant sample information objects. On the other hand a system analyst should be able to make a general model out of the sample information objects such that the model describes all other samples. Section 3 discusses these competencies in more detail.

The *effectiveness* of this way of working depends on how well its participants can accomplish their share, i.e. (1) how well can a domain expert provide a domain description, (2) how well can a domain expert validate a paraphrased description, (3) how well can a system analyst map sentences onto concepts, and (4) how well can a system analyst evaluate a validation. At least one iteration of the modeling loop is required, a bound on the maximal number of iterations is not available in this simple model. Usually modeling techniques tend to focus on modeling concepts and an associated tooling but less on the process of modeling, i.e. the way of working. To minimize and to control the number of iterations, this way of working requires methods that support this highly interactive process between domain experts and system analysts. Furthermore, guarantees for the quality of this process and the required competencies of the participants involved during RE are insufficiently addressed by most common methods.

In this paper focus is on the IM process, its quality and the required competencies of the participants in this process. Section 2 describes the IM process in more detail. The competencies of the participants of the IM process is subject of discussion in section 3. In section 4 the correctness of this way of working is verified and sketched.

2 The Information Modeling Process

In order to make a more fundamental statement about the quality of IM, we describe in figure 1 the modeling process in more depth by further elaborating the activities of *elicitation*, *modeling* and *validation*. We will also make more explicit what elements can be distinguished in a *formal specification*. In the next section we will make explicit what competencies are required from its participants, and use these competencies to verify this process.

The processes represented by the arrows labelled 4 upto 8 are suitable for automation support. A formal theory for these processes and corresponding models is elaborated in [13] and may be used as the underlying framework for building a supporting tool.

2.1 Elicitation

The stage *elicitation* is refined in figure 1 into the following substages (the numbers refer to the numbers in the figure):

1. Collect significant information objects from the application domain.
2. Verbalize these information objects in a common language.
3. Reformulate the initial specification into a unifying format.

The communication in the UoD may employ all kinds of information objects, for example text, graphics, etc. However, a textual description serves as a unifying format for all different media. Therefore the so-called *principle of universal linguistic expressibility* ([14]) is a presupposition for this modeling process:

All relevant information can and must be made explicit in a verbal way.

The way of working in this phase may benefit from linguistic tools, see [11], for example to detect similarities and ambiguities between sentences.

Independent of what modeling technique used, the (sentences in the) initial specification are to be reformulated in accordance with some unifying formatting strategy, leading to the informal specification. At this point the modeling process may start, during which the involvement of the domain expert is not required.

Only few conceptual modeling techniques provide the domain expert and system analyst with clues and rules which can be applied on the initial specification, such that the resulting informal specification can actually be used in the modeling process. Examples of such modeling techniques are NIAM ([15]), Object-Role Modeling ([16]) and PSM ([17]).

2.2 Modeling

The intention of the modeling phase is to transform an informal specification into a formal specification. This phase can be decomposed into the following substages (see figure 1):

4. Discover significant modeling concepts (syntactical categories) and their relationships.
5. Match sentence structure on modeling concepts.

During the grammatical analysis, syntactical variation is reduced to some syntactical normal form. The next step is to abstract from the sentence structures within the normal form specification and to match these sentence structures onto concepts of the modeling technique used. On the one hand, this step results in a structural framework (the conceptual model) and the rules for handling this framework (the constraints). Furthermore, the formal specification describes how these concepts can be addressed. Without loss of generality, we can state that this part of the specification contains a *lexicon* and rules to describe compound concepts (the *grammar rules*). As a side product, a *sample population* is obtained and put at the disposal of the *validation* activity.

The elementary and structure sentences of the normal form specification provide a simple and effective handle for obtaining the underlying conceptual model of so-called *Snapshot Information Systems* (see e.g. [18]), i.e. information systems where only the current state of the UoD is relevant. However, even though these informal specifications are an important aid in modeling information systems, they are still too poorly structured. One of the missing elements is the order and history of events. The mutual order of the sentences in an informal specification is lost, the analyst has to reconstruct this order. Other missing structural UoD properties are for instance related to the associates involved in events, and the role in which they are involved (see for example [19]).

2.3 Validation and Verification

The validation phase is further refined in figure 1 into the following stages:

6. Produce by paraphrases a textual description of the conceptual model using the information grammar and the lexicon.
7. Validate the textual description by comparing this description with the informal specification.
8. Check formal specification for internal consistency, e.g., check model for flexibility.

Once the information grammar is obtained as part of the formal specification, this grammar may be used to communicate the derived models to the domain experts for validation purposes. Two ways of validating the analysis models are considered:

1. producing a textual description of (part of) an analysis model using the information grammar.
2. obtaining a parser from the information grammar which provides the domain expert with the ability to check whether sentences of the expert language are captured by the (current version of the) information grammar. Note that this will be very useful when prototyping is used.

For an example of how to construct such a validation mechanism, see [20].

3 Information Modeling Competencies

The previous section describes IM from a process point of view. In this section, we discuss the actors and investigate the competencies that are required to provoke the intended behavior. Note that the processes labelled 4, 5, 6 and 8 have associated a single actor, i.e. the system analyst, while the other processes (labelled 1, 2, 3 and 7) have more actors. We will not consider any algorithmic synchronization aspects of cooperation between actors. Whether domain expert or system analyst actually are teams, is also not considered.

There are some interesting issues that are not discussed in this paper. For instance, social skills, such as negotiating and communicating, are out of scope. How the competencies are measured, enforced in practice, and how to check if they are applied, is also not discussed.

3.1 Domain Experts

A first base skill for a domain expert is the *completeness base skill*:

D-1. *Domain experts can provide a complete set of information objects.*

As obvious as this skill might seem, its impact is rather high: the skill is the foundation for correctness of the information system and provides insight into

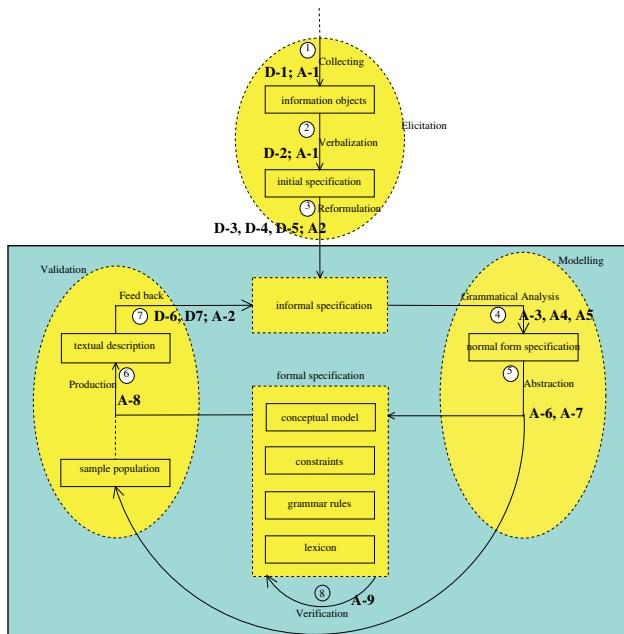


Fig. 1. Overview of competencies

the requirements for those who communicate the UoD to the system analyst during step 1. Furthermore, the skill is required to fulfill the *completion principle* as introduced in section 1. A second foundation for correctness, which also is introduced in section 1, is the *falsification principle*, which states that the derived formal specification is deemed to be correct if it does not conflict with the informal specification. To support this principle system analysts should have suitable skills, which are the topic of the next subsection.

The next step, labelled with verbalization, is the composition of a proper description of the information objects. It requires the domain expert to be familiar with the background of these objects, and being capable to describe them in every aspect. This is the purpose of the *provision base skill*:

D-2. *Domain experts can provide any number of significant sample sentences in relation to relevant information objects.*

This competence is a bottleneck for verbalization. As this base skill does not state that a domain expert can provide a *complete* set of significant examples by a single request from the system analyst, some more base skills that describe aspects of the *communication* between domain experts and system analysts are necessary.

A prerequisite for conceptual modeling is that sentences are elementary, i.e. not splittable without loss of information. As a system analyst is not assumed to be familiar with the semantics of sample sentences, it is up to the domain expert to judge about splitting sentences. This is expressed by the *splitting base skill*:

D-3. *Domain experts can split sample sentences into elementary sentences.*

A major advantage *and also* drawback of natural language is that there are usually several ways to express one particular event. For example, passive sentences can be reformulated into active sentences. By reformulating all sample sentences in a uniform way a system analyst can detect important syntactical categories during grammatical analysis of these reformulated sample sentences. The domain expert is responsible for reformulating the sample sentences, which is expressed by the *normalization base skill*:

D-4. *Domain experts can reformulate sample sentences into a unifying format.*

In order to capture the dynamics of the UoD the sentences need to be ordered. As a result, the domain expert has to be able to order the sample sentences. This is captured in the *ordering base skill*:

D-5. *Domain experts can order the sample sentences according to the dynamics of the application domain.*

The skills D-3, D-4 and D-5 are essential for reformulation (step 2 of the IM process).

During the modeling process, the information grammar (and thus the conceptual model) is constructed in a number of steps. After each step a provisional

information grammar is obtained, which can be used in the subsequent steps to communicate with domain experts. First a description of the model so far can be presented to the domain experts for validation. In the second place, the system analyst may confront the domain expert with a sample state of the UoD for validation. The goal of the system analyst might be to detect a specific constraint or to explore a subtype hierarchy. This is based on the *validation base skill*:

D-6. *Domain experts can validate a description of their application domain.*

This skill is essential for the validation in step 7 of the IM process. During this step of the IM process, the system analyst may check completeness by suggesting new sample sentences to the domain expert. This is based on the *significance base skill*:

D-7. *Domain experts can judge the significance of a sample sentence.*

The performance of the analysis process and the quality of its result benefits from the capability of the domain expert to become familiar with the concepts of the modeling technique. This is expressed by the *conceptuality base skill*:

D-8. *Domain experts have certain abilities to think on an abstract level.*

In contrast with the former skills, this latter skill is not required but highly desirable. This skill has a positive effect on all steps of the requirements engineering process.

3.2 System Analysts

Besides studying the cognitive identity of domain experts it is necessary to investigate the *cognitive identity* of system analysts.

Domain experts tend to leave out (unconsciously) those aspects in the application domain which are experienced as generally known and daily routine. An inexperienced system analyst may overlook such aspects leading to discussions on the explicit aspects, which usually addresses exceptions rather than rules. In order to detect implicit knowledge, the system analyst should not take things for granted, but should rather start with a minimal bias with a clean slate. This is expressed by the *tabula rasa base skill*:

A-1. *System analysts can handle implicit knowledge.*

This base skill is most related to the effectivity of the modeling process and the quality of its result, and is imperative during steps 1 and 2.

A next skill for system analysts is the so-called *consistency base skill*:

A-2. *System analysts can validate a set of sample sentences for consistency.*

The system analyst will benefit from this base skill especially during step 3. A major step in a natural language based modeling process is the detection of certain important syntactical categories. Therefore a system analyst must be able to perform a (possibly partially automated) grammatical analysis of the sample sentences. The result of this grammatical analysis is a number of related syntactical categories (step 4). This leads to the *grammar base skill*:

A-3. *System analysts can perform a grammatical analysis on a set of sample sentences.*

A system analyst is expected (during step 4) to make abstractions from detailed information provided by domain experts. By having more instances of some sort of sentence, the system analyst will get an impression of the underlying *sentence structure* and its appearances. The sentence structures all together form the structure of the information grammar. This is addressed in the *abstraction base skill* rule, which states that system analysts should be able to abstract from the result of the grammatical analysis:

A-4. *System analysts can abstract sentence structures from a set of related syntactical categories.*

Then (during step 5) a system analyst must be able to match sentence structures found with the abstraction base skill with the concepts of a particular conceptual modeling technique. This is expressed by the *modeling base skill* rule:

A-5. *System analysts can match abstract sentence structures with concepts of a modeling technique.*

As abstraction of sentences is based on the recognition of concrete manifestations, the system analyst must be able to generate new sample sentences which are validated by the domain experts. This enables the system analyst to formulate hypotheses via sample sentences. This way the system analyst has a mechanism to check boundaries, leading to data model constraints. Being able to generate sample sentences is expressed by the *generation base skill*:

A-6. *System analysts can generate new sample sentences.*

For example the system expert might resolve a case of ambiguity by offering (in step 7) well suited sample sentences to the domain expert for validation. This base skill is related to the ability of the system analyst to get acquainted with nature of the application domain, i.e., the counterpart of base skill D-8 for the domain expert. Note that base skill A-6 is required to give execution to the falsification principle as introduced in section 1.

Finally, the system analyst is expected to control the quality of the analysis models against requirements of well-formedness, i.e. the system analyst must satisfy the *fundamental base skill*:

A-7. *System analysts can think on an abstract level.*

The skills presented for the system analysts focus on a natural language based modeling process. Of course, system analysts and *system designers* also need expertise for using and understanding modeling techniques. In [21] a conceptual framework is proposed for examining these types of expertise. The components of this framework are applied to each phase of the development process and used to provide guidelines for the level of expertise developers might strive to obtain.

4 Controlling Natural Language

As stated in [22], natural language is the vehicle of our thoughts and their communication. Since good communication between system analyst and domain expert is essential for obtaining the intended information system, the communication between these two partners should be in a common language. Consequently natural language can be seen as a basis for communication between these two partners. Preferably natural language is used in both the modeling process as well as the validation process. In practice, system analyst and domain expert will have gained some knowledge of each other's expertise, making the role of natural language less emphasized, moving informal specification towards formal specification.

For the modeling process, natural language has the potential to be a precise specification language *provided* it is used well. Whereas a formal specification can never capture the pragmatics of a system, an initial specification in natural language provides clear hints on the way the users wants to communicate in the future with the information system.

Since modeling can be seen as mapping natural language concepts onto modeling technique concepts, paraphrasing can be seen as the inverse mapping intended as a feedback mechanism. This feedback mechanism increases the possibilities for domain experts to *validate* the formal specification, see e.g. [20] and ([23]). Besides the purpose of validation, paraphrasing is also useful to (1) lower the conceptual barrier of the domain expert, (2) to ease the understanding of the conceptual modeling formalism for the domain expert and (3) to ease the understanding of the UoD for the system analyst.

Up to now we focussed on the positive aspects of the usage of natural language, but in practice there are not many people who can use natural language in a (1) complete, (2) non-verbose, (3) unambiguous, (4) consistent way, (5) expressed on a uniform level of abstraction. In the sequel of this section we will make it plausible how the base skills for domain experts and systems analysts can reduce the impact of the these disadvantages, as these are the main critical succes factors of IM, of which the efficiency and effectiveness is a direct consequence.

The completeness and provision base skills (D-1 and D-2) anticipate on the completeness problem of natural language specifications. Skill D-2 states that domain experts can provide any number of significant sample sentences based on these information objects. Assuming that each UoD can be described by a finite number of structurally different sample sentences, the probability of missing some sentence structure decreases with each new sample sentence generated by the domain expert. Furthermore, the system analyst may generate sample sentences for validation in order to test correctness and completeness aspects of the formal specification sofar. These interactions are triggered, controlled and guided by the system analyst, as stated in the tabula rasa base skill A-1 and the consistency base skill A-6, to aim at convergence.

Specifications in natural language tend to be verbose, hiding essentials in linguistic variety. Complex (verbose) sentences will be feed back to the domain

expert for splitting (splitting base skill D-3) and judging significance for the problem domain (significance base skill D-7). A natural language specification may also get verbose by exemplification, providing examples (instantiations) of the same sentence structure. The grammar base skill A-3 reflects the ability of the system analyst to recognize the underlying similarity whereas base axiom A-4 provides the ability to abstract from superfluous sample sentences. Furthermore, the ability of domain experts to reformulate sentences in a unifying format (base axiom D-4) and to order sample sentences (base axiom D-5) is also helpful to eliminate the woolliness of initial specifications.

An often raised problem of natural language usage is ambiguity, i.e. sentences with the same sentence structure yet having a different meaning. The system analyst should have a nose for detecting ambiguities. The consistency base skill A-2 provides the system analyst with the required quality. A typical clue comes from establishing peculiarities in instantiated sentences. In order to decide about a suspected ambiguity, the system analyst will offer the domain expert these sentences for validation (generation base skill A-6 and validation base skill D-6).

On the other hand, the system analyst may also wish to elicit further explanation from the domain expert by requesting alternative formulations or more sample sentences with respect to the suspected ambiguity (provision base skill D-2).

The consistency base skill A-2 guarantees that the system analyst is equipped with the ability to verify a natural language specification for consistency. Just like the entire conceptual modeling process, consistency checking of natural language specifications has an iterative character. Furthermore, consistency checking requires interaction with the domain expert, as a system analyst may have either a request for more sample sentences (provision base skill D-2), or a request to validate new sample sentences (generation base skill A-6, validation base skill D-6 and significance base skill D-7).

Sentences of a natural language specification are often on a mixed level of abstraction. As a system analyst has limited detail knowledge, and thus also limited knowledge at the instance level, a prerequisite for abstraction is typing of instances (grammar base skill A-3 and abstraction base skill A-4) and map these types on the concepts of a modeling technique (modeling base skill A-5). The analysis of instances within a sentence is in fact a form of *typing*, attributing types to each of its components. As an example of such a sentence consider: *The Rolling Stones record the song Paint It Black*. Some instances will be typed by the domain expert (the song *Paint It Black*) while others are untyped (*The Rolling Stones*). This may be resolved by applying a *type inference mechanism* to untyped instances (see [19]). Typed sentences can be presented to the domain expert for validation (base skill D-6).

Although a formal mathematical proof can not be provided the above line of reasoning makes it plausible that the disadvantages of natural language usage can be overcome if the participants in de the requirements engineering process do have the required skills.

Acknowledgements. The authors would like to thank Denis Verhoef and Marc Lankhorst for reviewing this paper and providing very useful comments.

References

1. Royce, W.: Managing the development of large software systems: Concepts and techniques. In: 9th International Conference on Software Engineering, Los Angeles, California, IEEE WESTCON (1970) 1–9
2. Booch, G., Jacobson, I., Rumbaugh, J.: The rational Unified Process, an Introduction. Addison-Wesley (2000)
3. Boehm, B.: A spiral model of software development and enhancement. *IEEE Computer* **21** (1988) 61–72
4. Boehm, B.: Cots integration: plug and pray? *IEEE Software* **32** (1999) 135–138
5. Linger, R.: Cleanroom process model. *IEEE Software* **11** (1994) 50–58
6. Aurum, A., Wohlin, C.: The fundamental nature of requirements engineering activities as a decision-making process. *Information and Software Technology* (2003)
7. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. In: 22nd International Conference on Software Engineering, Ireland, ACM (2000) 35–46
8. Sommerville, I.: Software Engineering. 6th edn. Addison-Wesley, Reading, Massachusetts (2001)
9. Pressman, R.: Software Engineering. 5th edn. McGraw-Hill, England (2000)
10. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley (1999)
11. Burg, J.: Linguistic Instruments In Requirements Engineering. PhD thesis, Free University, Amsterdam, The Netherlands (1996)
12. Bray, I.: An introduction to Requirements Engineering. Addison Wesley, Edinburg Gate, United Kingdom (2002)
13. Frederiks, P., Weide, T.v.d.: Deriving and paraphrasing information grammars using object-oriented analysis models. *Acta Informatica* **38** (2002) 437–488
14. Adriaans, P.: Language Learning from a Categorial Perspective. PhD thesis, University of Amsterdam, Amsterdam, The Netherlands (1992)
15. Nijssen, G., Halpin, T.: Conceptual Schema and Relational Database Design: a fact oriented approach. Prentice-Hall, Sydney, Australia (1989)
16. Halpin, T., Bloesch, A.: Data modeling in uml and orm: a comparison (1999)
17. Hofstede, A.t., Weide, T.v.d.: Expressiveness in conceptual data modelling. *Data & Knowledge Engineering* **10** (1993) 65–100
18. Date, C.: An Introduction to Data Base Systems. 7th edn. Addison-Wesley, Reading, Massachusetts (1999)
19. Bommel, P.v., Frederiks, P., Weide, T.v.d.: Object-Oriented Modeling based on Logbooks. *The Computer Journal* **39** (1996)
20. Derksen, C., Frederiks, P., Weide, T.v.d.: Paraphrasing as a Technique to Support Object-Oriented Analysis. In Riet, R.v.d., Burg, J., Vos, A.v.d., eds.: Proceedings of the Second Workshop on Applications of Natural Language to Databases (NLDB'96), Amsterdam, The Netherlands (1996) 28–39
21. Storey, V., Thompson, C., Ram, S.: Understanding database design expertise. *Data & Knowledge Engineering* **16** (1995) 97–124
22. Quine, W.: Word and object – Studies in communication. The Technology Press of the Massachusetts Institute of Technology, Cambridge, Massachusetts (1960)
23. Dalianis, H.: Aggregation in Natural Language Generation. *Journal of Computational Intelligence* **15** (1999) 384–414

Experimenting with Linguistic Tools for Conceptual Modelling: Quality of the Models and Critical Features

Nadzeya Kiyavitskaya¹, Nicola Zeni¹, Luisa Mich¹, and John Mylopoulos²

¹Department of Information and Communication Technologies, University of Trento
Via Sommarive 14, 38050, Povo, Trento, Italy

{nadzeya.kiyavitskaya, nicola.zeni, luisa.mich}@unitn.it,

²Department of Computer Science, University of Toronto
jm@cs.toronto.edu

Abstract. This paper presents the results of three experiments designed to assess the extent to which a Natural-Language Processing (NLP) tool improves the quality of conceptual models, specifically object-oriented ones. Our main experimental hypothesis is that the quality of a domain class model is higher if its development is supported by a NLP system. The tool used for the experiment – named NL-OOPS – extracts classes and associations from a knowledge base realized by a deep semantic analysis of a sample text. In our experiments, we had groups working with and without the tool, and then compared and evaluated the final class models they produced. The results of the experiments give insights on the state of the art of NL-based Computer Aided Software Engineering (CASE) tools and allow identifying important guidelines to improve their performance, highlighting which of the linguistic tasks are more critical to effectively support conceptual modelling.

1 Introduction

According to the results of a market research whose aim was to analyse the potential demand for a CASE tool integrating linguistic instruments as support for requirements analysis, 79% of requirements documents are couched in unrestricted NL. Also the majority of developers (64%) pointed out that the most useful thing to improve general efficiency in modelling user requirements would be a higher level of automation [20]. However, there is still no commercial NL-based CASE tool. There have been many attempts to develop tools that support requirements engineering since the '80s. The objective of this work was to evaluate how the NL-based CASE tools can support the modelling process, thereby speeding up requirements formalisation. This research makes use of linguistic techniques which were considered state-of-the-art at that time, although newer technologies have now been developed. In this work we present the results of a set of experiments designed to investigate the extent to which a state-of-the-art NLP tool that supports the semi-automatic construction of a conceptual model improves their quality. The tool used for the experiments – named NL-OOPS – extracts classes and associations from a knowledge base realized by a deep semantic analysis of a sample text [13]. In particular, NL-OOPS produces class models at different levels of detail by exploiting class hierarchies in the knowledge base of the

NLP system and marks ambiguities in the text [16], [17], [18]. In our experiments, we had groups and individuals working with and without the tool, and then compared and evaluated the final class models they produced. The results of the experiments give some insight on the state of the art of NL-based CASE tools and identify some important parameters for improving their performances.

Section 2 of the paper presents related research projects that use linguistic tools of different complexity to support conceptual modelling. Section 3 describes the main features of NL-OOPS and basic knowledge upon which the NLP system is built. Section 4 outlines the stages of the experiments and contains an evaluation of the models produced, focusing on the effect that NL-OOPS had on their quality. The concluding section summarises the findings of the experiment and describes directions for future research.

2 Theoretical Background

The use of linguistic tools to support conceptual modelling was proposed by the large number of studies. We will report here only some of the most important related works to illustrate the efforts toward the development of NL-based CASE tools. In the early 1980's, Abbott [1] proposed an approach to program design based on linguistic analysis of informal strategies written in English. This approach was further developed by Booch [4], who proposed a syntactic analysis of the problem description. Saeki, Horai, and Enomoto [27] were the first to use linguistic tools for requirements analysis. Dunn and Orlowska [10] described a NL interpreter for the construction of NIAM conceptual schemas. Another relevant work was the expert system ALECSI [6] that used a semantic network to represent domain knowledge [25]. Along similar lines, Cockburn [7] investigated the application of linguistic metaphors to object-oriented design. One of the first attempts to apply automated tools to requirements analysis is proposed in [9]. Goldin and Berry [14] introduce the approach for finding abstractions in NL text using signal processing methods. The COLOR-X project attempts to minimize the participation of the user in the job of extracting classes and relationships from the text [5]. A method whose objective was to eliminate ambiguity in NL requirements by using a Controlled Language is presented in [22]. The approach described in [3] for producing interactively conceptual models of NL requirements was to use a domain dictionary and a set of fuzzy-logic rules. Among the recent projects, an interactive method and a prototype – LIDA - is described in [24]. However, most of the text analysis remains a manual process. Another research area connected both with the conceptual modelling and NLP is investigation of the quality of requirements' language [17], [23]. In particular, some authors focused on the support of writing requirements [2].

In this context, NL-OOPS¹ - the tool used in the experiments and described in the next section - presents a higher degree of automation. It is based on a large NLP system [13], that made it possible to produce completely automatically a draft of a conceptual model starting from narrative text in unrestricted NL (English) [16].

¹ <http://nl-oops.cs.unitn.it>

3 The NL-OOPS Tool

There are two complementary approaches to develop a tool for extraction from textual descriptions the elements necessary to design and build conceptual models. The first limits the use of NL to a subset that can be analysed syntactically. Various dialects of "Structured English" do that. The drawback of this method is that it will not work for real text. The second approach adopts NLP systems capable of understanding the content of documents by means of a semantic, or deep, analysis. The obvious advantage of such systems is their application to arbitrary NL text. Moreover, such systems can cope with ambiguities in syntax, semantics, pragmatics, or discourse. Clearly, compared to the first category, they are much more complex, require further research, and have a limited scope.

NL-OOPS is an NL-based CASE prototype. It was founded on LOLITA (Large-scale Object-based Language Interactor, Translator and Analyser) NLP system, which includes all the functions for analysis of NL: morphology, parsing (1500-rules grammar), semantic and pragmatic analysis, inference, and generation [13]. The knowledge base of the system consists of a semantic network, which contains about 150,000 nodes. Thus LOLITA is among the largest implemented NLP systems. Documents in English are analysed by LOLITA and their content is stored in its knowledge base, adding new nodes to its semantic network. NL-OOPS prototype implements an algorithm for the extraction of classes and associations from the semantic network of LOLITA. The NL-OOPS's interface consists of three frames [15]. First one contains the text being analysed, the second frame gives a partial representation of the SemNet structures used by LOLITA for the analysis of the document. After running the modelling module, the third frame gives a version of the class model. The tool can export intermediate results to a Word file or a Java source file; traceability function allows the user to check what nodes were created for a given sentence. The nodes browser of NL-OOPS makes available further information related to a specific node, e.g. the hierarchies in which it is involved.

4 The Experiments

Our main experimental hypothesis was that when model development is supported by a NLP-based tool, the quality of the domain class model is higher and the design productivity increases. Consequently the goal of the experiments was to confirm or refute this assumption and then to identify the features and the linguistic tasks that effective NL-based CASE system should include.

4.1 Realization of the Experiments

In each experiment, we assigned a software requirements document to the participants and we asked them to develop in a given time a class domain model, identifying classes, associations, multiplicity, attributes and methods. Half of the participants were supported by the NL-OOPS tool. They were also given some training in the use

of NL-OOPS functionalities: changing the threshold for the algorithm to produce models at different levels of detail; viewing the list of candidate classes, and navigating the nodes browser of the knowledge base. The chosen class model could then be deployed in a java file, which was reverse engineered into PoseidonCE² or Rational Rose³ class diagrams. Both these means create the list of classes and the analyst has only to drag them in the diagram, and then to check and complete the diagram. Before the experiment, we administered a short questionnaire to assess the experience of the participants. To compare the results of the experiments, we used the same requirements text in all the experiments. In particular, the text named Softcom [26], deals with a problem that requires some familiarity with judged sports. The language is quite simple, but also realistic. It contains all the typical features for requirements text: use of the passive voice, etc. The second case study named Library [11] had a level of difficulty similar to that of the Softcom case. Both texts are cited in [15]. The first two experiments involved couples of analysts. The first experiment focused on the quality of the models obtained with and without NL-OOPS; in the second, participants were asked to save the diagrams at fixed time intervals, to obtain data also about productivity. In the third experiment, each analyst worked alone developing two models for two different problem statements, one with the tool and one without it. For the first two experiments participants were undergraduate students and their competence in object-oriented conceptual modelling was comparable to that of junior analysts. For the last experiment participants were PhD students with higher competence.

The classes suggested by NL-OOPS with different thresholds are given for Softcom in and Library in Table 1 and 2, respectively (threshold influences on the depth of LOLITA's semantic network hierarchies). These classes constitute the main input for the analysts working with NL-OOPS. To interpret the results we refer to the class models proposed with the problem sources [11], [26] (last column in table 1 and 2). The names of the classes in the reference models are in bold. This choice was made to minimize the subjectivity in the evaluation of the models produced by the participants. We calculated recall (R, counts the number of correct identified classes divided by total number of correct classes), precision (P, counts the number of correct identified classes divided by total number of classes), and the F-measure (combines R and P) to evaluate the performance for the class identification task [29].

The models proposed by NL-OOPS do not contain classes; they instead present in the list of candidate classes. In the first two cases two classes are indicated: entity (worker), entity (announcer), corresponding to ambiguity in the text. In the first case, entity was introduced by the NLP system for the sentence "Working from stations, the judges can score many competitions": it cannot be automatically assumed that the subject of working is "judges". The second class results from an analysis of the sentence "In a particular competition, competitors receive a number which is announced and used to split them into groups", where the subject of announces is unknown. The use of the node browser of NL-OOPS allows to go back to the original phrase to determine whether it gives the information necessary for the model.

² Gentleware: <http://www.gentleware.com>

³ IBM – Rational Software: www.rational.com

Table 1. Classes identified by NL-OOPS: SoftCom case

NLOOP-1 (12)	NL-OOPS-2 (10)	NL-OOPS-3 (5)	Reference classes (11)
Competition			Competition
Competition (subclass of Competition in SemNet)			
Competitor	Competitor	Competitor	Competitor
Entity (worker)*	Entity (worker)		
Entity (announcer)	Entity (announcer)		
			Figure (styles, routines)
Group	Group	Group	
High	High		
Judge	Judge	Judge	Judge
			League
Meeting	Meeting		Meeting
Number	Number	Number	
Score	Score	Score	Score
			Season
			Station
			Team
			Trial
Softcom	Softcom		
R=45.5%; P=41.7%, F-measure =43.5%	R=36.4%; P=40.0%, F-measure = 38.1%	R=27.3%; P=60.0%, F-measure =37.5%	R _{avg} =36.4%; P _{avg} =47.2% F-measure _{avg} =39.7%

* Word in parenthesis corresponds to the actual meaning of concept shown implicitly in network

For the Library case, the measures of the class identification task are higher than for the SoftCom case. However, the quality of the models produced by NL-OOPS is reduced by the presence of classes due to unresolved anaphoric references (“It”, “Entity”, “Pair”), or to ambiguity in the sentences. For example, the subject in sentence “The reservation is cancelled when the borrower check out the book or magazine or through an explicit cancelling procedure” is omitted. Another spurious class is “Murder”, which was introduced by LOLITA as subject of an event related to the “remove”-action (due to the absence of domain knowledge).

4.2 Analysis of the Results

Evaluating the quality of the models is a subjective process. The experience gained from the experiments and the analysis of the literature about quality of conceptual model,⁴ helped us to define a schema to support their evaluation. The schema take into

⁴ There are only few papers about this topic; see for example, [21], [28].

Table 2. Classes identified by NL-OOPS: Library case

NLOOPs-1 (17)	NL-OOPS-2 (10)	NL-OOPS-3 (7)	Reference classes (7)
Book	Book	Book	Book
Borrower	Borrower	Borrower	Borrower
Person			
Copy			<i>Item (Book Copy; Magazine Copy)*</i>
Employee	Employee		
Entity (delete, update, create)	Entity (delete, update, create)	Entity (delete, update, create)	
Entity (cancel)			
Entity (register)			
Software_System	Software_System	Soft-ware_System	
It (Library)	It (Library)	It (Library)	
Entity (Library)			
Library	Library	Library	
Loan			<i>Loan</i>
Magazine	Magazine	Magazine	<i>Magazine</i>
Murderer (remove)			
Purchase			
Reservation	Reservation		<i>Reservation</i>
Thing (superclass of Book and Magazine)	<i>Thing (superclass of Book and Magazine)</i>		
Pair (superclass of Book and Magazine)			
Title			<i>Title (Book Title; Magazine Title)*</i>
R = 100%, P= 41.2% F-measure = 58.4%	R=57.1-71.4%, P=40.0-50.0%** F-measure =47.0-58.8%	R=42.9%, P=42.9%, F-measure =42.9%	R _{avg} =66.7%-71.4%; P _{avg} =41.4%-44.7%; F-measure _{avg} =49.4%-53.4%

* The hierarchies for *Copy* and *Title* represent two alternatives used to evaluate the model developed by the students

** The maximum values were calculated including the class *Thing*

account the criteria related to both external and internal quality (considered in [28]), evaluating:

- the content of the model i.e. semantic quality: how much and how deep the model represents the problem,
- the form of the model i.e. syntactic quality (proper and extensive use of UML notation, operating all the variety of UML expressions such of aggregation, inheritance, etc.),
- the quantity of identified items (in class model: number of classes, attributes, operations, associations, and hierarchies).

Each of these criteria reflects a particular aspect of model quality. To evaluate each one we assign a scale with a range from 0 (lowest mark) to 5 (highest mark). The overall quality of the models is measured basing on mixed approach:

- the application of the quality schema,
- the evaluation by two experts, that assessed the models as they usually do for their students' projects.

For the class identification task, which is a part of the conceptual model design task the quality was evaluated calculating recall, precision, and the F-measure.

First Experiment. In the first preliminary experiment the group of twelve students was split into six subgroups [19]. Each group had access to a PC with Microsoft Office 2000 while carrying out the experiment. Three groups worked with NL-OOPS. The length of the experiment was 90 minutes. For the six diagrams produced, two groups used PowerPoint, one used Excel, and all groups working without NL-OOPS chose Word. The results of the identification task are given in Table 3.

Table 3. Class identification

	1 tool	2 tool	3 tool	1	2	3
Recall	72.7%	54.5%	81.8%	100.0%	100.0%	81.8%
Precision	88.9%	66.7%	69.2%	78.6%	68.8%	90.0%
F-measure	80.0%	60.0%	75.0%	88.0%	81.5%	85.7%

To evaluate the overall quality of class diagrams, we asked two experts to mark and comment on the solutions proposed by the different groups (table 4). The experts judged the best model to be the one produced by group 5, in which two of the students had used UML for real projects. So, if on this basis, it was excluded, in order to have comparable level of groups, the best model would be one developed with the support of NL-OOPS. Considering these results with those in table 1, the tool seemed to have an inertial effect that on one hand led to the tacit acceptance of classes (e.g., group); on the other hand it resulted in the failure to introduce some indispensable classes (e.g., season, team). From the analysis of the feedbacks given by the participants some considerations emerged: (a) those who used NL-OOPS would have preferred more training; (b) each group that used NL-OOPS would prefer to have a tool to design the diagrams, while groups working without the tool did not voice this preference. All these considerations were used for the realisation of the subsequent experiments.

Table 4. Experts' evaluation of overall model quality

Groups	Quality
1 tool	pretty good
2 tool	low
3 tool	good
4	pretty good
5	good
6	low

Second Experiment. We repeated the experiment later with involving students. Participants were divided into five groups: two of them used NL-OOPS. The participants had access to Poseidon CE. We asked them to produce the model of the domain classes for the problem assigned. The length of the experiment was set for 1 hour. As we wanted to obtain also information regarding the productivity of conceptual modelling supported by linguistic tools, we asked the students to save every fifteen minutes screen shot of their model. The performances related to the class identification task are summarised in the table 5, we report average recall, precision and F-measure for both groups.

Table 5. Class identification

	15'	30'	45'	60'
Recall tool	45.5%	69.7%	75.7%	75.7%
	50.0%	59.1%	63.6%	81.8%
Precision tool	33.4%	71.3%	74.2%	76.4%
	52.6%	60.3%	72.9%	73.3%
F-measure tool	38.5%	66.2%	73.5%	74.7%
	50.9%	59.2%	67.3%	75.8%

Marks and comments on the overall quality made by two experts are given in table 6.

Table 6. Expert evaluation of overall model quality

Groups	Quality
1	low
2	pretty good
3	good
4 tool	pretty good
5 tool	low

The experts judged the best model to be the one produced by group 3 which participants (according to the questionnaire) used UML for real projects. The application of the quality schema described in section 4.2 gives the results in table 7.

Table 7. Overall Quality

Time	Content				Form				Items				Total			
	15'	30'	45'	60'	15'	30'	45'	60'	15'	30'	45'	60'	15'	30'	45'	60'
no tool	0.0	1.7	2.0	3.3	0.0	1.7	2.1	3.7	1.1	2.6	2.9	3.9	0.4	2.0	2.3	3.6
With tool	3.0	3.3	2.5	3.5	3.0	3.2	2.5	2.8	2.2	2.2	3.4	4.1	2.8	2.9	2.8	3.5

Third Experiment. In the third experiment we made some more changes. First of all, the participants worked individually. They had to deal with two different problem statements of comparable difficulty, with and without the NL-OOPS prototype. We set the length of the experiment to 20 minutes for each case. As in the second experiment we decide to collect progressive results, so we asked them to save the model in an intermediate file after the first 10 minutes. The results for the class identification task are presented in the table 8. We should comment that even though the experts chose the requirement texts of comparable level, for the linguistic analysis there was the difference. For instance, Library case turned to be more difficult for the NLP system to understand because it contains many anaphors (table 1-2).

Table 8. Class identification

Parameter	Case	10'		20'	
		Recall	tool	Precision	tool
Recall	softcom	51.5%		49.6%	70.9%
	library	47.6%			53.6%
	softcom	47.6%		59.5%	53.6%
	library	71.4%			71.4%
Precision	softcom	85.2%		75.9%	72.5%
	library	66.7%			51.5%
	softcom	66.7%		58.3%	51.5%
	library	50.0%			50.2%
F-measure	softcom	64.2%		59.9%	71.7%
	library	55.6%			52.5%
	softcom	55.6%		57.2%	52.5%
	library	58.8%			59.0%
					55.7%

We can assume here existence of some inertial effect because the users tend to keep all the candidate classes provided by NL-OOPS without getting rid of the fake classes (“it”, “thing”, “entity”, etc.).

The application of the quality schema described in section 4.2 gives the results in table 9. Marks and comments on the overall quality made by two experts are given in table 10. In this experiment both quality and productivity had been improved thanks to the support of the NL-OOPS tool, even though the participants were pessimistic about using such kind of linguistic instrument.

Table 9. Overall Quality

Parameter	Case	10'		20'	
Content tool	softcom	1.4	1.3	3.9	3.5
	library	1.1		3	
	softcom	2.1	2.1	3.5	3.9
	library	2		4.2	
Form tool	softcom	1.5	1.4	3.8	3.4
	library	1.3		3	
	softcom	2.4	2.5	4.2	4.5
	library	2.6		4.9	
Items tool	softcom	0.7	1	3	3.1
	library	1.2		3.2	
	softcom	1.7	2	4.1	4.4
	library	2.2		4.7	
Total tool	softcom	1.2	1.2	3.6	3.3
	library	1.2		3.1	
	softcom	2.1	2.2	3.9	4.3
	library	2.3		4.6	

Table 10. Expert evaluation of overall model quality

Person	Evaluation			
	Time	10'	10' tool	20'
1	low	pretty good	low	good
2	*	-	pretty good	good
3	low	good	low	good
4	low	pretty good	low	good
5	low	pretty good	pretty good	good
6	-	-	pretty good	good
7**	-	-	-	-
8	-	low	pretty good	low
9	low	good	low	good
10	low	-	low	low

*Grey cells correspond to Softcom

**Person 7 violated the rules of experiment, so the data cannot be considered as correct

5 Conclusions

The empirical results from the three experiments neither confirm nor refute the initial hypothesis of this paper that the quality of a domain class model is higher if its development is supported by a NLP system. There is some evidence, however, that model quality is better for NLP tool users early on during the modelling process. See the results of the third experiment at 10 minutes in Table 9. As to the impact of a NLP tool on productivity, the results of the experiments are uniformly inconclusive, but there is some evidence in Tables 7 and 9 that users work faster when supported by the tool. We interpret these results to mean that at initial steps the tool is helpful in

speeding up the work, but by the end of the process, the advantage is lost because the users have to go into details of the text anyway to verify the correctness of list of classes and to derive other elements of the class diagrams. The prototype was in some ways misused, as users were not able to take advantage of all the functionality provided by the system. Apparently, the groups working with the tool used only the initial draft of the class model and only part of the list of the candidate classes produced by the tool. A user did not go deep into the details of the semantic network constructed by the system and focused his/her attention only on the final list of the most probable classes candidates. To avoid this effect, NL-OOPS should have better integration of the different types of information it generates with the diagram visualization tools. On the methodological level, the quality evaluation schema and the approach we adopted for the experiments described in this paper for the evaluation of NL-OOPS can be used to evaluate the output produced by any case tool designed to support the modelling process. Other lessons learned from the experiments regarding features for an effective NLP-based CASE tool, include:

- the knowledge base produced by the linguistic analysis must be presentable in a user-understandable form,
- the most and least probable class and relationship candidates should be highlighted, to help the user modify the final model, either by extending it with other classes or by deleting irrelevant ones,
- the tool should be interactive to allow the analyst to resolve ambiguities and reflect these changes in the semantic representation immediately.

In general terms, the experiments confirm that, given the state of the art for NLP systems, heavyweight tools are not effective in supporting conceptual model construction. Instead, it makes sense to adopt lightweight linguistic tools that can be tailored to particular linguistic analysis tasks and scale up. Moreover, linguistic analysis may be more useful for large textual documents that need to be analysed quickly (but not necessarily very accurately), rather than short documents that need to be analysed carefully. We will be focusing our future research towards this direction.

References

1. Abbott R., “Program Design by Informal English Descriptions”, Comm. of the ACM, 26 (11): 882-894, 1983
2. Aguilera C., Berry D. M., “The Use of a Repeated Phrase Finder in Requirements Extraction”. *Journal of Systems and Software*, 13: 209-230, 1990
3. Ambriola V., Gervasi V. “An Environment for Cooperative Construction of Natural-Language Requirements Bases”, In Proc. 8th Conf. on SWE Environments, IEEE, 1997
4. Booch G., “Object Oriented Development”, *IEEE Trans. on SE*, 12 (2): 211-221, 1986
5. Burg, J.F.M., “*Linguistic Instruments in Requirements Engineering*”, IOS Press, 1996
6. Cauvet C., Proix C., Rolland C., “ALECSI: An Expert System for Requirements Engineering”, *Conf. on Advanced Information Systems Engineering - CAiSE*, Norway, p.31-49, May 1991
7. Cockburn A., “Using NL as a Metaphoric Basis for Object-Oriented Modelling and Programming”, IBM Technical Report, TR-36.0002, 1992
8. Cockburn A., “Using NL as a metaphoric base for OO”, In Proc. Conf. on Object Oriented Programming Systems Languages and Applications Archive, Canada, p.187-189, 1993

9. Cordes D., CarverD., "An Object-Based Requirements Modelling Method", *Journal of the American Society for Information Science*, 43 (1): 62-71, 1992
10. Dunn L., Orlowska M. "A NL interpreter for construction of conceptual schemas", *In Proc. 2nd Conf. on Advanced IS Engineering - CAiSE'90*, LNCS 436, p.371-386, Springer-Verlag, 1990
11. Eriksson H-E., Penker M., *UML Toolkit*, John Wiley, New York, 1998
12. Ingalls Daniel H. H., "The Smalltalk-76 programming system design and implementation", *In Proc. 5th ACM Symp. on Principles of programming lang.*, Tucson, Jan. 23-25, p.9-16, 1978
13. Gariglano R., Urbanowicz A., Nettleton D.J., "Description of the LOLITA system as Used in MUC 7". *In Proc. Conf. MUC 7*, 1997
14. Goldin L., Berry D. M.: "AbstFinder, A Prototype NL Text Abstraction Finder for Use in Requirements Elicitation", *In Proc. 1st Int. Conf. on RE* Colorado Springs, CO, IEEE, 1994
15. Kiyavitskaya N., Zeni N., Mich L., Mylopoulos J., "NLP-Based Requirements Modeling: Experiments on the Quality of the models", Technical Report DIT, University of Trento, 2004
16. Mich L., "NL-OOPS: From NL to OO Requirements using the NLP System LOLITA", *Jour. of NL Engineering*, Cambridge University Press, 2 (2): 161-187, 1996
17. Mich L., Gariglano R., "Ambiguity measures in Requirements Engineering", *In Proc. Int. Conf. on SW - Theory and Practice - ICS2000, 16th IFIP Cong.*, Beijing, p.9-48, 21-25 Aug. 2000
18. Mich, L. and Gariglano, R. "NL-OOPS: A Requirements Analysis tool based on NL Processing". *In Proc. 3rd Int. Conf. on Data Mining 2002*, Bologna, 25-27 Sep. 2002, p.321-330
19. Mich L., Mylopoulos J., Zeni N., "Improving the Quality of Conceptual Models with NLP Tools: An Experiment", Tech. Report DIT, University of Trento, 2002
20. Mich L., Franch M., Novi Inverardi P.L., "Requirements Analysis using linguistic tools: Results of an On-line Survey", *Journal of Requirements Engineering*, Springer-Verlag, Oct. 2003
21. Moody D.L., Shanks G.G. "What Makes a Good Data Model? A Framework for Evaluating and Improving the Quality of ER Models". *Australian Computer Journal* 30(3): 97-110, 1998
22. Osborne M., MacNish C.K., "Processing NL software requirement specifications". *In Proc. 2nd IEEE Int. Conf. on Requirements Engineering (ICRE'96)*, IEEE, p. 229-236, 1996
23. Fabbrini F., Fusani M., Gnesi S., Lami G. "Quality Evaluation of Software Requirements Specifications", *In Proc. 13th International SW Quality Week Conference*, 2000
24. Overmyer S.P., Lavoie B., Rambow O., "Conceptual Modelling through Linguistic Analysis Using LIDA", *In Proc. 23rd Int. Conf. on SW engineering (ICSE 2001)*, Jul. 2001, p.401-410
25. Rolland C., Proix C., "A NL approach for requirements engineering". *In Proc. 4th International Conference CAiSE 1992*, p. 257-277.
26. Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W., *Object-Oriented Modelling and Design*, Prentice-Hall, 1991
27. Saeki M., Horai H., Enomoto H., "Software Development Process from NL Specification". *In Proc. 11th Int. Conf. on SW Engineering*, Pittsburgh, PE, Mar. 1989, p.64-73.
28. Teeuw W.B., Van den Berg H., "On the Quality of Conceptual Models", *In Proc. Work. Behavioral models and design transformations: Issues and opportunities in conceptual modelling - ER'97*, USA, Nov. 6-7, 1997,
<http://osm7.cs.byu.edu/ER97/workshop4/tvdb.html>
29. van Rijsbergen C.J., *Information Retrieval*, Butterworths, 1979.

Language Resources and Tools for Supporting the System Engineering Process

V.O. Onditi, P. Rayson, B. Ransom, D. Ramduny, Ian Sommerville, and A. Dix

Computing Department, Lancaster University, UK, LA1 4YR
`{onditi, paul, bjr, devina, is, dixa}@comp.lancs.ac.uk`

Abstract. This paper discusses an approach to tracking decisions made in meetings from documentation such as minutes and storing them in such a way as to support efficient retrieval. Decisions are intended to inform future actions and activities but over time the decisions and their rationale are often forgotten. Our studies have found that decisions, their rationale and the relationships between decisions are frequently not recorded or often buried deeply in text. Consequently, subsequent decisions are delayed or misinformed. Recently, there has been an increased interest in the preservation of group knowledge invested in the development of systems and a corresponding increase in the technologies used for capturing the information. This results in huge information repositories. However, the existing support for processing the vast amount of information is insufficient. We seek to uncover and track decisions in order to make them readily available for future use, thus reducing rework.

1 Introduction

Solutions to systems engineering¹ problems are products of collaborative work over a period of time. Several people with varied expertise and experience invest their knowledge in the product. During the product's development, several decisions are made. Some are about the product, others about the process. These decisions can broadly be classified into two: system decisions and process decisions. System decisions deal with the technical aspects of a system such as its features, architecture, reliability, safety, usability etc. Process decisions include responsibility delegation, milestones, follow-on actions, schedules, even budget. We are interested particularly in process decisions that require actions to be taken in order to progress a project. This work forms part of our research on the Tracker² project where we seek to understand the nature of decisions in teams and organisations; in particular the way past decisions are acted on, referred to, forgotten about and otherwise function as part of long term organizational activity.

Meetings are important activities in collaborative work. They represent activities within a process. There is a history of previous meetings, a constellation of concepts and documents that are brought into the meeting, which often evolves as a result of the meeting, and are taken into work activities afterwards [1]. The meetings drive

¹ The development of software systems

² <http://www.comp.lancs.ac.uk/computing/research/cseg/projects/tracker/>

processes, which in turn are driven by the outcomes of these processes. The outcome of a meeting is communicated through minutes. The standard practice in many organisations is to circulate minutes to the meetings participants (the list of attendees, apologies and absentees). However, some decisions require the attention of people who aren't members of the meetings. Consequently, decisions must be extracted from minutes and communicated to the external audience. The failure to meet this responsibility can lead to misinformed decisions or a delay in subsequent decisions being made.

We argue that finding, associating and communicating decisions can be automated. Natural Language Processing (NLP) techniques can be used for identifying action verbs (actions) and subjects (agents) in sentences. Anecdotal evidence shows that minutes are often circulated late, just before a subsequent meeting. Then it is too late to remember what transpired in the previous meeting or to extract decisions and communicate them to external audience. In addition, few people read the minutes in good time even if the minutes are circulated early. Over time the minutes become too voluminous and overwhelming to analyse manually. As a result participants go into meetings ill prepared leading to unsatisfactory participation.

In the requirements engineering field, language resources have been used to identify systems decisions (candidate features) by picking out verbs from requirements documents [9, 10]. In the same way, nouns can be identified. This is important because nouns are considered candidate objects in Object-Oriented technology. Furthermore requirements can be categorised based on modal verbs into mandatory, less obligatory etc [10].

A similar approach can be used for identifying actions³ and finding associations between actions. Automatic extraction of actions and relationships between them can provide the means for interpreting process decisions. This provides similar support to the system engineering process as the identification of verbs and nouns for system decisions. Also, actions provide a means for tracking the progress of a process and may also be used to estimate individual contribution in a collaborative work.

2 Tokenisation and Linguistic Annotation

Before we can analyse text we need to tokenise it. Tokenisation involves breaking a document into constituent parts based on the structure and style of the document. The tokenised text can be represented as plain text or in XML format. XML based representation provides a better way to separate content from its meta data (including formatting information). Thus, the two concerns, content and rendering, can be addressed separately. Meta data is used for analysing content and formatting information for rendering the content. Many NLP tools require plain text and represent semantic or part-of speech information as plain text. For example, using C7 part-of speech (pos) and semantic tag set [6, 9], **Item 1** would be represented in plain text as follows:

Item/ id="1.1"/ pos="NN 1"/ sem="O2" 1/ id="1.2"/ pos="MC1"/ sem="N 1".

³ A statement which specifies what should be done and who should do it.

In XML, the previous example would be represented as

```
<w id="2.1" pos="NN1" sem="O2">Item</ w >
<w id="2.2" pos="MC1" sem="N1">l</ w >
```

Where w = word, id = the ordinal number of a word in a sentence, pos = part of speech and sem = semantic category. We will adopt the latter representation. For the purpose of this paper, we will discuss how to convert a Microsoft Word document internally represented in rich text format (.rtf), or word document (.doc) to XML.

2.1 Document Style and Structure Based Tokenisation and Annotation

We are developing a decision capture tool (Fig. 4.) which accepts a document in rich text format (.rtf) or Microsoft Word document format (.doc) as input and returns an XML document. Typically, word documents consist of sections, paragraphs, sentences and words and use different formatting styles such as *italics*, underline and **bold** face. Such information represents hidden meaning, for example a paragraph break may indicate the introduction of a new issue or a different viewpoint. Similarly non-identical font style between two adjacent paragraphs suggests the introduction of a different item or viewpoint. The tool uses such information to introduce annotations to a document for example paragraphs have `<paragraph>` tag around them (Fig. 2.). In addition, indicator words such as agenda (issue), minute (statement) or action are added to enclose the document units' annotations. Indicator words and phrases have been shown to be useful in automatic summarisation [8].

A structural tagger uses the template, (Fig. 1.) to introduce appropriate annotations. The element paragraph consists of a series of sentences, represented by element 's' and each sentence contains a series of words, represented by element 'w'. Each word has attributes 'id' (ordinal number), 'sem' (semantic) are 'pos' (part-of-speech).

The structural tagger breaks the text into paragraph units. The linguistic tagger (see Linguistic Annotation) adds the smaller units such as sentences and words.

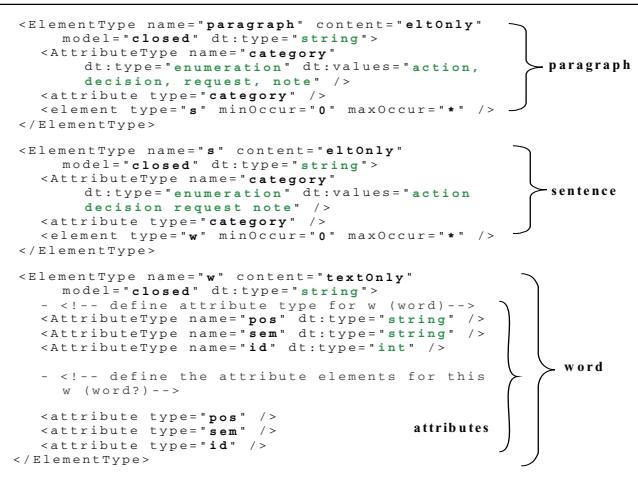


Fig. 1. Structural layers of a word document

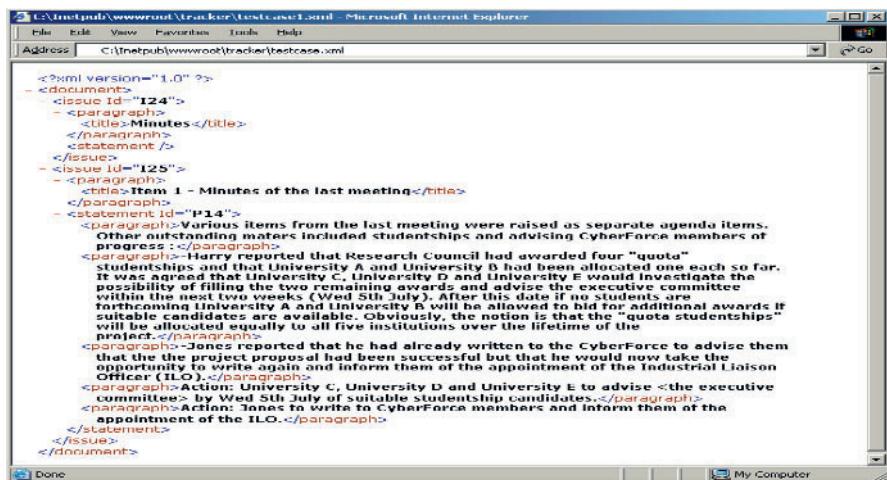


Fig. 2. Identifying and annotating text constituencies

Fig. 2. shows a document after structural annotations are introduced. The first paragraph is identified as an issue because it is bold face in the original document. Similarly, the second paragraph is identified as an issue because the first line is in bold face. However, the third paragraph is not an issue. Since it comes immediately after an issue paragraph, it is identified as a minute statement. It is logical to argue that a paragraph immediately after an agenda item and with a different formatting style is a minute statement.

The converted (structurally annotated) document is then passed through a linguistic tagger. The linguistic tagger further breaks the document into sentences and words and introduces meta data such as ordinal number (id), part-of-speech (pos), and semantic (sem) category (see examples under Linguistic Annotation). The output is a document that conforms to the template in Fig. 1.

2.2 Linguistic Annotation

We use two existing NLP tools, namely CLAWS⁴ and Semantic Analyser⁵ that employ a hybrid approach consisting of rule-based and probabilistic NLP techniques. CLAWS [7] is an important tool for processing English language text. It uses a statistical hidden Markov model technique and a rule-based component to identify the parts-of-speech of words to an accuracy of 97-98%. The Semantic Analyser [10] uses the POS-tagged text to assign semantic annotations that represent the general semantic field of words from a lexicon of single words and an idiom list of multi-word combinations (e.g. ‘as a rule’). These language resources contain approximately 61,400 words and idioms and classify them according to a hierarchy of semantic classes.

⁴ <http://www.comp.lancs.ac.uk/ucrel/claws/>

⁵ <http://www.comp.lancs.ac.uk/ucrel/usas/>

The tools were trained on large corpora of free text that had been analysed and ‘tagged’ with each word’s syntactic or semantic category. Extremely large corpora have been compiled, for instance, the British National Corpus consists of approximately 100 million words [2]. For some levels of analysis, notably POS annotation, probabilistic NLP tools have been able to achieve very high levels of accuracy and robustness unconstrained by the richness of language used or the volume of documentation.

Such techniques extract interesting properties of text that a human user can combine and use to infer meanings. Evidence from other domains suggests that such tools can effectively support the analysis of large documentary sources. For example, NLP tools were used to confirm the results of a painstaking manual discourse analysis of doctor-patient interaction [11] and they also revealed information that had not been discovered manually.

3 Analysing Content

Words are shells within which meaning lies. When we break the shell, the inner meaning emerges. At the surface is the literal (surface) meaning; deeper within is figurative and metaphorical meaning. For example, the Kiswahili word *safari* invokes different meanings to different people. Literally, *safari* is a Swahili word for journey, a long and arduous one. However, it invokes a different meaning amongst the Western societies than the Swahili speaking communities of East Africa. In the West, *safari* is almost synonymous with touring and game viewing. In this perspective, it is a happy outing.

3.1 Surface Analysis

Surface analysis involves the identification of indicator words in a domain and constraining their meaning within the domain. Using indicator words, we can determine specific elements of information in a document. For example, in a student’s answer booklet, the word Answer is used to suggest a solution even though we are aware that some are incorrect. If taken literally, all the worked solutions would be correct. In this case, the use of the word Answer serves to identify and separate solutions from each other. In documents such as minutes of meetings, the same approach can be used for the analysis of elements such as agenda items (issues) and minutes (solution statement). In a minutes document, agenda items are often marked with the tag ‘agenda’ followed by an ordinal number. Similarly, minutes are marked with a minute tag followed by a number.

Our approach incorporates document structure and style to distinguish ordinary numbered items from minute elements (Fig. 2.). The body of a minutes’ document consists of agendum-minute pair. We have observed that these sections are formatted differently. A new element starts on a new paragraph and a new paragraph with a different formatting from the previous one suggests a new element. For example, agenda items frequently have formatting such as italics, bold face or underline. Using such styles, in combination with the surface meaning of the indicator words or word phrases, we can retrieve agenda items and minute statements. This approach is suit-

able within a domain. If taken outside the domain, the meaning of indicator words may change considerably.

While our approach coped fairly well with the set of minutes used in the study (Table 1), it is not our claim that it will work in every situation. A standard document template is desirable. We are developing a template for taking minutes that can be used with our tool.

3.2 Semantic (Deep) Analysis

Since surface implies something exists underneath, we require deeper analysis to discover the meaning of sentences in a certain context. Such meaning is represented in the part-of-speech and semantic categories of Natural Language Processing. Analysis at this level can reveal more information, e.g. two sentences can be shown to be similar even though different words or syntax is used. For example, the following sentences carry the same meaning but use different words and syntax:

- a) Site XYZ should develop test cases for module A1
- b) Site XYZ to create test cases for module A1

In these examples, the words develop and create, or their inflected forms have semantic category values A2.1 and A2.2 respectively. A2.1 means affect: modify or change, A2.2 means affect: cause⁶. These are both subdivision of the affect category. Similarly, words with the same semantic categories (semantic chains) could be used in places of develop or create.

- c) <s><w id="2.1" pos="NN1" sem="M7">Site</w> <w id="2.2" pos="FO" sem="Z3c">XYZ</w> <w id="2.3" pos="VM" sem="S6+">should</w><w id="2.4" pos="VVI" sem="A2.1+">develop</w> <w id="2.5" pos="NN1" sem="P1">test</w> <w id="2.6" pos="NN2" sem="A4.1">cases</w> <w id="2.7" pos="IF" sem="Z5">for</w> <w id="2.8" pos="NN1" sem="P1">module</w> <w id="2.9" pos="FO" sem="Z2">A1</w> <w id="2.10" pos=". " sem="PUNC"></w> </s>
- d) <s><w id="3.1" pos="NN1" sem="M7">Site</w> <w id="3.2" pos="FO" sem="Z3c">XYZ</w> <w id="3.3" pos="TO" sem="Z5">to</w> <w id="3.4" pos="VVI" sem="A2.2">create</w> <w id="3.5" pos="RP" sem="N2|i1.2.2">out</w><w id="3.5" pos="NN1" sem="P1">test</w> <w id="3.6" pos="NN2" sem="A4.1">cases</w> <w id="3.7" pos="IF" sem="Z5">for</w> <w id="3.8" pos="NN1" sem="P1">module</w> <w id="3.9" pos="FO" sem="Z2">A1</w> <w id="3.10" pos=". " sem="PUNC"></w> </s>

The sentences have a subject (Site XYZ) and an object (test cases). The subject can be syntactically arranged so that it appears at the head (a & b) or the tail of a sentence. In examples (a and b), the sentences are of the form:

subject + infinitive verb (object)

We studied several minutes' documents to discover semantic and part-of-speech patterns that represented actions statements. Below are some examples from the minutes.

- e) Jeff to provide Activity 2 document in template format before Monday 26 June.
- f) Harry and Eileen to provide some text on qualitative work for Activity 4 to Jeff by Monday morning 26 June.

We came to the hypothesis that actions that will occur in the future identify agent or agents (who) and the action (what). Also, we observed that the sentences contained a modal verb or a function word 'to'. Further, agents and action-describing verbs are connected by modal verbs or the function word 'to'. We argue that a simple future

⁶ <http://www.comp.lancs.ac.uk/ucrel/usas/semtags.txt>

action sentence exists in three parts: a noun or noun phrase, a function word ‘to’ or a modal verb and an action verb or verb phrase. The noun phrase may be a proper noun such as the name of a person e.g. Tom, a geographical name such as London or common nouns such as names of groups of people e.g. partners, members etc. A verb expresses the action (what will happen). It may also contain a subordinate clause, for example a constraint.

Davidson’s [5] treatment of action verbs (verbs that describes what someone did, is doing or will do) as containing a place for singular terms or variables is consistent with our observation. For example, the statement “Amundsen flew to the North Pole” is represented as $(\exists x) (\text{Flew}(Amundsen, \text{North Pole}, x))$. Davidson called the representation a logical form of action sentence. Though Davidson considered actions that had occurred, the logical form also applies to future actions. For example, the action sentences (e) and (f) above can be expressed in Davidson’s logical form as: $(\exists x) (\text{Provide}(Jeff, \text{Activity 2 document in template format}, x))$ and $(\exists x) (\text{Provide}(Harry \text{ and Eileen, some text on qualitative work for Activity 4}, x) \& \text{To}(Jeff, x))$ respectively. The strength of Davidson’s logical form of action sentence is it’s extensibility as illustrated in example (f) above.

From these results, we developed the following protocol for extracting action sentences.

Noun + function word ‘to’/modal verb + infinitive verbp1
The above protocol can be represented as a template as:

semantic=“agent” word=“to”, POS=“infinitive verb”

where agent matches the noun phrase and the verb phrase matches both the infinitive verb and the subordinate clause.

It is important to note that the noun, modal verb and action verb must occur in a particular order in any action sentence. The sentences have a leading noun or noun phrase followed by a modal verb or a function word ‘to’ and an infinitive verb. If the order changes, the meaning is altered. For example, the modal verb ‘can’ conveys the sense of ability to do something e.g. “I can drive”. It can also be used to pose a question – “Can you drive?”. Similarly, other modal verbs such as could, shall, should, must, may, might, will, would, ought to can be used in the same way. The senses that modal verbs convey include ability, advice, expectation, necessity, request, question and possibility.

In the C7 tag set⁷, all nouns have a part-of-speech category that starts with N. For example, NN represents common noun, NP proper noun etc. Using such information, we are able to identify nouns in sentences. Similarly, verbs have part-of-speech values that start with V. For example VB⁸ represents the verb ‘be’ with all its inflected forms, VD* represents the verb ‘do’ and its inflected forms. The function word ‘to’ has a part-of-speech value of TO and semantic categories of either X7 or S6 which indicate planning, choosing, obligation or necessity. It is possible to map part-of-speech and semantic categories onto the protocol (p1) above as follows:

N* + (TO/ VM) + V* = action statementp2

The protocol (p2) retrieved not only all the actions that we identified by intuitive interpretation of the same text but also identified others. Fig. 4. shows a list of actions in a contrast background in a web browser.

⁷ <http://www.comp.lancs.ac.uk/ucrel/claws7tags.html>

⁸ one or more letters

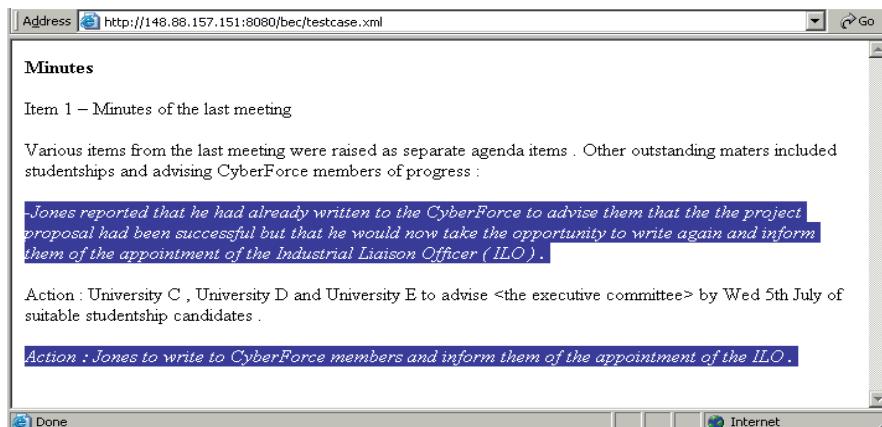


Fig. 3. Actions in a contrasting background

3.3 Analysing Sentences for Semantic Relationships

Issues are related in different ways – the relationship between two issues could be in containment i.e. an issue is a sub-issue of another. Actions could be related by the fact that they were made in the same meeting or, are being implemented by the same person. Still, they could be related by content i.e. they talk about the same thing. We can determine that two sentences talk about the same thing by comparing their content words (nouns, verbs, adjectives and adverbs). In examples (a) and (b) in section 3.2 (Semantic (Deep) Analysis) above, the meaning of the two sentences is determined by the sets of words {site, XYZ, should, develop, test, cases, module, A1} and {site, XYZ, create, test, cases, module, A1}

The relationship between these two sentences is calculated by comparing the frequencies of the sense of each word in the first set to the second set. We use the sense of each word deliberately because nearly every sentence in English can be written in two or more different ways. The sense is represented in the semantic category. Thus, the sets above translates to {M7, Z3c, S6+, A2.1, P1, A4.1, P1, Z2} and {M7, Z3c, VVI, A2.2, A4.1, P1, Z2} respectively. The correspondence between sentence (a) and sentence (b) is $|A \cap B| / |B|$ or $|A \cap B| / |A|$ whichever is the smallest, where A is the set of semantic values for sentence (a) and B is the set of semantic values for sentence (b). Thus the similarity between sentence (a) and sentence (b) is 0.75. The smallest value is selected to take into account the difference in lengths between the two sentences. For example, if set B has only one member, and that member is also a member of set A, then we would claim that the correspondence between (a) and (b) is 1.0. Such a claim would be inaccurate because of the difference in the length of the sentences.

The correspondence values measure the semantic association between two sentences. As a result, it is possible to track and retrieve the relationships amongst issues and actions. The relationship is based on the content of a document, not its metadata. Fig. 4 shows an interface for retrieving related information.

4 Representing Issues, Actions, and Relationships on a Database

We developed a tool utilizing a simple relational database that ties the unstructured minutes and the structured information identified through the methods discussed in sections above. The tokenisation and annotation technique, discussed in one of the previous sections, structures documents which maps onto a database structure. The database consists of issues, statements about the issues, actions, associations and agents. Issues, actions and minutes statements are bookmarked. Association between issues and actions are also represented on the database (Fig. 4. bottom frame).

The database can also be used to provide services such as communication. The tool also integrates a communication service. The service periodically interrogates the database for new actions and posts them to agents. These communications are recorded and could be tracked.

In meetings, actions can be tracked by receiving and recording reports from agents or their representatives. Also, the outcomes of these actions can be recorded. This can be particularly important for providing rationales for actions and reasons for decisions taken.

5 Tool Evaluation

We tested the protocol (p2) against a set of twelve minutes taken from four different organisations. The organisations (A, B, C and D) are arranged along the first column (Table 1). The sets are numbered from 1 to 3 for each organisation. Organisation A minutes were used in developing the document template (Fig. 1.). In the table, Recall is the ratio of the number of relevant actions retrieved to the total number of relevant actions in the document. Precision is the ratio of the number of relevant actions retrieved to the total number of relevant and irrelevant actions retrieved. Rel. (relevant) is the number of relevant actions in a minutes document. Ret. (retrieved) is the number of actions returned by the protocol (p2) in section 3.2 (Semantic (Deep) Analysis). RelRet. (Relevant Retrieved) is the number of relevant actions retrieved. Thus if there are 20 actions in a minutes document and the protocol returns 16 items out of which 10 are relevant then Precision = 50% and Recall = 62.5%. The average precisions for the four organisations are 85%, 27%, 56% and 80% respectively. While the average recalls are 80%, 43%, 95% and 95% respectively.

Generally, the recall rate is good, an overall average of 78.4%. Precision stands at about 62.18% overall. This is attributed to different factors ranging from personalised styles of writing minutes to grammatical mistakes. In some of the minutes' sets studied, statements which are not actions linguistically were annotated as actions while some statements were disjointed and therefore could not properly form a complete sentence. Since our tool depends on linguistic annotator to overcome grammatical mistakes, there is a knock-on effect on our tool.

Table 1. Decision Capture Tool's Information Retrieval efficiency results

	A	B	C	D	E	F	G
	Set	RelRet	Rel	Ret	Recall	Precision	
1							
2	A	1	70	82	81	85.37%	86.42%
3		2	31	35	40	88.57%	77.50%
4		3	33	50	36	66.00%	91.67%
5	B	1	4	4	9	100.00%	44.44%
6		2	3	17	13	17.65%	23.08%
7		3	2	16	15	12.50%	13.33%
8	C	1	1	1	2	100.00%	50.00%
9		2	23	27	29	85.19%	79.31%
10		3	4	4	10	100.00%	40.00%
11	D	1	9	9	9	100.00%	100.00%
12		2	14	15	19	93.33%	73.68%
13		3	12	13	18	92.31%	66.67%
14	Average				78.41%	62.18%	

We are also measuring the correspondence amongst the actions. However, since it uses the same principle as identifying actions, we are confident that the result will not vary significantly. We predict that the technique could also be applied to spoken discourse but with reduced accuracy. Our preliminary results on the application of the technique on spoken discourse indeed confirm this prediction.

6 Rendering Analysed Text

In section 2 (Tokenisation and Linguistic Analysis), we argued for the separation of content and presentation. In this section, we discuss how to render the processed document. Extracted information can be rendered in two ways: as a separate list of actions or highlighted ‘in-text’. As a separate list, the context is eliminated to enable people to concentrate on actions. For example, Fig. 4. shows a list of actions extracted from minutes’ documents. This could be useful during reviews of previous actions in subsequent meetings. Other elements of minutes such as agenda and minute statements can also be viewed separately from the context.

More importantly, actions can be viewed in text (Fig. 3.) thereby preserving the context. To help with readability, actions are shown in a contrasting background. A style sheet⁹ is used for rendering the analysed document on a web browser. The style sheet is based on the minutes template described in section 2 (Tokenisation and Linguistic Annotation). The template applies different formatting styles to different parts of a document and shows action elements in a contrasting background. Using the web interface, it is possible to browse information on the database and jump directly to where the element appears in text.

In Fig. 4., the left pane shows available functionalities and categories of information. The information is organized around the categories based on the agenda items and the dates on which the agenda items were discussed. Fig. 4. could be used for posting minutes to Tracker server. The tracker service runs at scheduled interval to process minutes document. This involves issue, action and minute statement identification, extraction and posting to a database. It also involves the calculation of relationship amongst actions and issues. The calculated values are stored on the database.

⁹ <http://www.comp.lancs.ac.uk/computing/research/cseg/projects/tracker/test.xls>

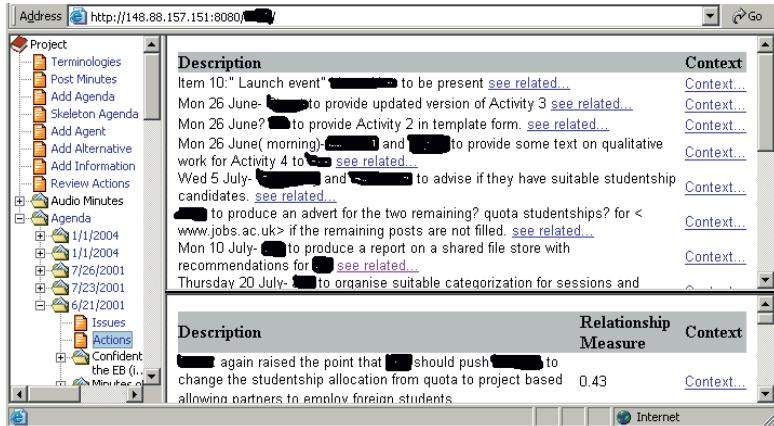


Fig. 4. Web interface for browsing decision elements

The tool (Fig. 4.) can also be used in different situations: pre-, in- and post-meeting to provide services such as eliciting agenda items, adding agents and reviewing action to capture their outcomes. On the top right frame, details about the information category selected from the left pane are displayed. For example, Fig. 4. lists actions from the minutes' of 21/06/2001. The bottom, right frame lists the related actions from the one selected in the top right frame.¹⁰ The tool also supports a query facility. A user can start with a query which then returns the list in Fig. 4. Through the context link, it is possible to jump to the location where the text occurred.

7 Related Work

Few meeting capture tools are available off-the shelf and more research is ongoing. Different dimensions of decision capture are under investigation; some emphasise capture, others representation. Still others emphasise retrieval. But all the three issues are intertwined to the extent that they cannot be separately addressed. The area of capture seems to be widely researched as evident in tools [3, 4]. However, the areas of representation and retrieval remain a great challenge [1]. It is our view that more research into techniques for unobtrusive structuring is needed.

8 Conclusion

In this paper, we have demonstrated language resources and tools for capturing actions and issues from minutes to support system engineering process. The use of language resources and tools has enabled us to extract actions from minutes documents. Although this is useful on its own, we do not regard this as a standalone technique but

¹⁰ for the purposes of anonymity, some information is blanked out in Fig. 4.

one to be used in conjunction with audio and video minutes. It forms part of a broader goal to create tools to facilitate the tracking and re-use of decisions.

We have also noticed that precision and recall are inversely proportional. Improving one impacts negatively on the other. We think that information is more difficult to find than to identify, thus we have worked to improve recall.

Acknowledgement. The Tracker project is supported under the EPSRC Systems Integration programme in the UK, project number GR/R12183/01.

References

1. Albert, S., Shum, S. B., Maarten, S., Jeff, C., Beatrix, Z., Charles, P., Wilfred, D., David, H., John, D., Enrico, M. & G., L.: Compendium: Making Meetings into Knowledge Events, Knowledge Technologies, March 4-7, Austin TX (2001)
2. Aston G, Burnard L.: The BNC Handbook: Exploring the British National Corpus with SARA, Edinburgh University Press (1998).
3. Chiu, P., Boreczky, J., Gergensohn, A., and Kimber, D.: LiteMinutes: An Internet-Based System for Multimedia Minutes, Proceedings of Tenth World Wide Web Conference, ACM Press (2001) pp. 140-149.
4. Chiu, P. et. al.: NoteLook: Taking notes in meetings with digital video and ink, Proceedings of ACM Multimedia, ACM Press, (1999) pp. 149-158.
5. Davidson, D.: Essays on Actions & Events, Oxford University Press (1980) pp. 105-122
6. Garside, R., Leech, G., and McEnery A.: Corpus Annotation, Longman, London and New York (1997a).
7. Garside R, Smith N. A: Hybrid Grammatical Tagger: CLAWS4, in Garside R, Leech G, McEnery, A. (eds.) Corpus Annotation: Linguistic Information from Computer Text Corpora, Longman (1997b).
8. Hovy, E. and Lin, C.: Automated Text Summarisation In SUMMARIST, in Mani I and Maybury M. (eds.) Advances in Automatic Text Summarisation, MIT Press (1999) pp 81-97
9. Rayson, P., and Wilson, A.: The ACAMRIT semantic annotation system: progress report, in L. J. Evett, and T. G. Rose (eds) Language Engineering for Document Analysis and Recognition, LEDAR, AISB96 Workshop proceedings, Brighton, England (1996) pp 13-20.
10. Sawyer, P., Rayson, P., and Garside, R.: REVERE: support for requirements synthesis from documents, Information Systems Frontiers Journal, 4(3), Kluwer, Netherlands (2002) pp 343-353.
11. Thomas J. et. al.: Methodologies for Studying a Corpus of Doctor-Patient Interaction, in Thomas J, Short M. (eds.) Using Corpora for Language Research, Longman (1996).

A Linguistics-Based Approach for Use Case Driven Analysis Using Goal and Scenario Authoring*

Jintae Kim¹, Sooyong Park¹, and Vijayan Sugumaran²

¹ Department of Computer Science, Sogang University
Sinsu-Dong, Mapo-gu, Seoul, 121-742, Republic of Korea
`{canon, sypark}@sogang.ac.kr`

² Department of Decision and Information Sciences
Oakland University, Rochester, MI 48309, USA
`sugumara@oakland.edu`

Abstract. Although Use Case driven analysis has been widely used in requirements analysis, it does not facilitate effective requirements elicitation or provide rationale for the various artifacts that get generated. On the other hand, goal and scenario based approach is considered to be effective for elicitation but it does not lead to use cases. This paper discusses how to combine goal and scenario based requirements elicitation technique with use case driven analysis using natural language concepts. In our proposed approach, four levels of goals, scenario authoring rules, and linguistic techniques have been developed to identify use cases from text based goal and scenario descriptions. The artifacts resulting from our approach could be used as input to a use case diagramming tool to automate the process of use case diagram generation.

1 Introduction

Use case driven analysis (UCDA) has been one of the most popular analysis methods in Requirements engineering. Use case driven analysis helps to cope with the complexity of the requirements analysis process. By identifying and then independently analyzing different use cases, we may focus on one narrow aspect of the system usage at a time. Since the idea of UCDA is simple, and the use case descriptions are based on natural concepts that can be found in the problem domain, the customers and the end users can actively participate in requirements analysis. Consequently, developers can learn more about the potential users, their actual needs, and their typical behavior [1]. However, the lack of support for a systematic requirements elicitation process is probably one of the main drawbacks of UCDA. This lack of elicitation guidance in UCDA sometimes results in an ad hoc set of use cases without any underlying rationale. On the other hand, if one knows the origin of each use case, one could capture requirements through UCDA more completely. Our current research addresses the issue of the lack of elicitation support in UCDA by using goal and scenario modeling. Thus, the objective of this paper is to develop an approach for use case analysis that

* This work was supported by the National Research Laboratory (NRL) Program of the Ministry of Science and Technology of Korea and University IT Research Center Project.

makes use of goal and scenario authoring techniques using natural language processing concepts.

Goal modeling is an effective way to identify requirements [2] [11]. The main emphasis of goal driven approaches is that the rationale for developing a system is to be found outside the system itself – in the enterprise in which the system shall function [3] [4]. A goal provides a rationale for requirements, i.e., a requirement exists because of some underlying goal, which provides the basis for it [11] [12] [13]. Recently, some proposals have been made to integrate goals and scenarios together. The impetus for this is the notion that by capturing examples and illustrations, scenarios can help people in reasoning about complex systems [5].

One of the early proposals that combine a use case with other concepts is that of Cockburn [6]. It suggests the use of goals to structure use cases by connecting every action in a scenario to a goal. However, Cockburn's approach is just concerned with the description of scenarios in a use case. Ralyté [7] integrated scenario based techniques into existing methods. This led to some enhancement of use case modeling within the OOSE method. However, Ralyté's approach does not provide any rationale for identifying use cases, i.e., it cannot reflect where the use cases come from. Finally, these approaches do not support both requirements elicitation and requirements analysis. Little is known about supporting the elicitation process in UCDA.

In this paper, we present an approach that supports deriving use cases from goal modeling through authoring scenarios. Especially, a linguistics-based technique for scenario authoring, and goal modeling with different levels is proposed to complement the elicitation process within UCDA. Our aim is to provide a supplementary elicitation process, which helps a human engineer to analyze the system with UCDA through goal and scenario modeling. Figure 1 depicts an overview of our approach.

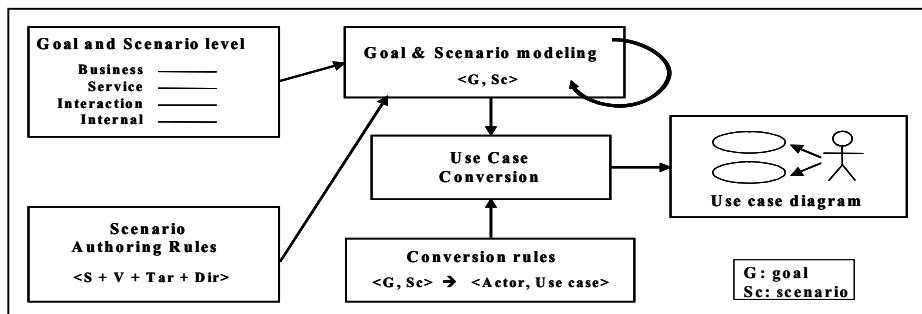


Fig. 1. Overview of our approach

Our approach consists of the following two main activities: a) goals and scenarios are generated for each goal and scenario level, namely, Business, Service, Interaction, and Internal, and b) conversion rules are used to transform the scenarios into a use case diagram. When scenarios achieving a goal are authored, they are generated using scenario authoring rules outlined in this paper. Goal and scenario authoring activity is highly iterative that results in a successively refined set of goals and scenarios. The output of the higher level becomes input to the immediate lower level, thus yielding a hierarchy of goals and scenarios with various levels. The following three characteristics exemplify our approach and contribute to the objective of systematically gener-

ating use case diagrams from goal and scenario modeling. Essentially, our approach bridges the gap between the goal and scenario modeling research and the traditional use case driven analysis within requirements engineering.

The first characteristic of our approach is the *abstraction level of goal and scenario* (AL). Based on [8] [9], we classify *<Goal, Scenario>* pairs into business, service, system interaction, and system internal levels. As a result, the goals and scenarios are organized in a four level abstraction hierarchy. This hierarchy-based approach helps separate concerns in requirements elicitation and also refining a goal.

The second characteristic is the scenario-authoring rule. Scenarios are very familiar to users. However, scenario authoring is ambiguous to users and it is not easy to articulate requirements through scenarios because there are no rules to author scenarios. Scenario authoring rules help elicit requirements from goals and they could be very useful to users and developers alike.

The third characteristic of our approach is the concept of conversion rules for mapping goal and scenario to actor and use case. These rules guide the user to elicit actors and use cases. There are two key notions embodied in the conversion rules. The first idea is that each goal at the interaction level is mapped to a use case because the scenarios, which achieve the goal at this level, are represented as ‘the interactions between the system and its agents. This definition is similar to use case’s definition. The second idea is that each scenario achieving a goal at this level describes the flow of events in a use case. Therefore, this idea signifies that the rationale for use cases is founded on goals, which are derived through scenarios.

The remainder of the paper is structured as follows. The abstraction levels for goal and scenario modeling are described in the next section. Goal and scenario modeling based on linguistic techniques is presented in Section 3. Section 4 deals with use case conversion rules. The concluding section sums up the essential properties of our approach and the contributions of this paper.

2 Four Abstraction Levels of Goal and Scenario

Four abstraction levels of goal and scenarios (AL) help separate concerns in requirements elicitation. Prior research has proved the usefulness of multiple levels of abstractions by applying the approach to the ELEKTRA real world case [10]. In this paper, we propose a four level abstraction hierarchy, organized as business, service, interaction, and internal level. Goal modeling is accompanied by scenarios corresponding to each of the abstraction levels. A goal is created at each level and scenarios are generated to achieve the goal. This is a convenient way to elicit requirements through goal modeling because these levels make it possible to refine the goals [8] [11]. Four abstraction levels of goal and scenario modeling are discussed below.

2.1 Business Level

The aim of the business level is to identify the ultimate purpose of a system. At this level, the overall system goal is specified by the organization or a particular user. For example, the business goal ‘Improve the services provided to our bank customers’ is an overall goal set up by the banking organization.

2.2 Service Level

The aim of the service level is to identify the services that a system should provide to an organization and their rationale. At this level, several alternative architectures of services are postulated and evaluated. All of them correspond to a given business goal. Goals and scenarios in service level are represented as a pair $\langle G, Sc \rangle$ where G is a design goal and Sc is a service scenario. A design goal expresses one possible manner of fulfilling the business goal. For example, the design goal ‘Cash withdraw’ is one possible way of satisfying the business goal. A service scenario describes the flow of services among agents, which are necessary to fulfill the design goal. For example, the service scenario ‘The customer withdraws cash from the ATM’ implements the design goal ‘Cash withdraw’.

2.3 Interaction Level

At the system interaction level the focus is on the interactions between the system and its agents. Goals and scenarios at this level are represented as a pair $\langle G, Sc \rangle$ where G is a service goal and Sc is an interaction scenario. These interactions are required to achieve the services assigned to the system at the service level. The service goal ‘Withdraw cash from ATM’ expresses a manner of providing a service. Interaction scenario describes the flow of interactions between the system and agents. For example, the interaction scenario ‘The ATM receives the amount from user’ implements the service goal.

2.4 Internal Level

The internal level focuses on what the system needs to perform the interactions selected at the system interaction level. The ‘what’ is expressed in terms of internal system actions that involve system objects but may require external objects such as other systems. At this level, goals and scenarios are represented as a pair $\langle G, Sc \rangle$ where G is an interaction goal and Sc is an internal scenario. For example, ‘Deliver cash to the user’ is an interaction goal. The associated internal scenario describes the flow of interactions among the system objects to fulfill the interaction goal.

3 Goal and Scenario Modeling Using Linguistics

This section discusses goal and scenario modeling specific to each abstraction level using linguistic concepts. The notions of goal and scenario are briefly described. Then, the scenario-authoring rules are discussed.

3.1 The Concept of Goal and Scenario

A goal is defined as “something that some stakeholder hopes to achieve in the future” [4] [8] [11]. A scenario is “a possible behavior limited to a set of purposeful interac-

tions taking place among several agents” [14]. The scenarios capture real requirements since they describe real situations or concrete behaviors, and goals can be achieved through the execution of scenarios. Thus, scenarios have their goals, and typically, goals are achieved by scenarios. In other words, just as goals can help in scenario discovery, scenarios can also help in goal discovery. As each individual goal is discovered, a scenario can be authored for it. Once a scenario has been authored, it can be explored to yield further goals [3] [11].

3.2 Scenario Authoring Model

As stated earlier, one can think of scenario-authoring rules as a set of guidelines that help generate scenarios. They are based on the linguistic techniques. First, we briefly discuss the scenario structure and then the scenario authoring rules. Our scenario structure is an extension of Rolland’s approach [8] and due to space limitation we do not provide an extensive discussion on scenario structure from [8].

3.2.1 The Scenario Structure

Since a goal is intentional and a scenario is operational by nature, a goal is achieved by one or more scenarios. Scenario structure is a template, which enables us to describe scenarios for the goal. Figure 2 shows our scenario structure is composed of several components that are analogous to parts of speech, i.e., elements of a sentence.

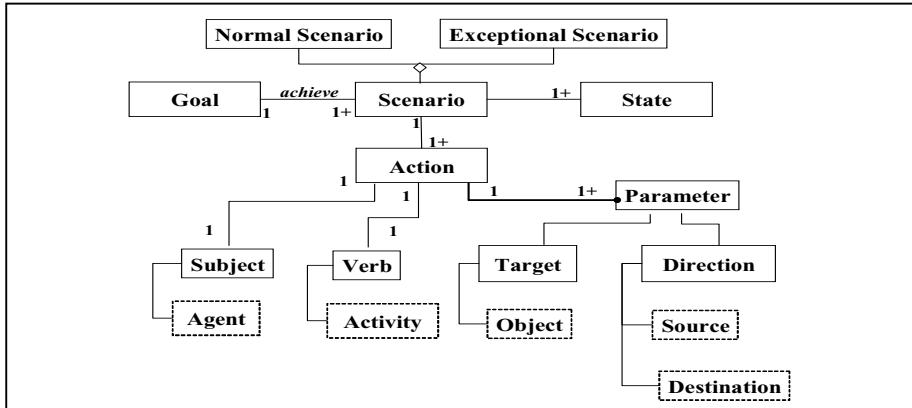


Fig. 2. The scenario structure

A scenario is associated with a subject, a verb and one or more parameters (multiplicity is shown by a black dot). It is expressed as a clause with a subject, a main verb, and several parameters, where each parameter plays a different role with respect to the verb. There are two types of parameters (shown in the dash boxes). The main components of the scenario structure are described with an ATM example.

The agent (Agent) is responsible for a given goal and implements an activity as an active entity. An actor or the system in the sentence can be the agent: for example in

the scenario, ‘*(The user)_{Agent} inserts card into ATM*’. The activity (*Act*) is the main verb. It shows one step in the sequence of scenarios. The object (*Obj*) is conceptual or a physical entity which is changed by activities: for example, ‘*The Customer deposits cash_{Obj} to the ATM*’. An object has several properties such as ownership, place, status, amount, etc., which can be changed by activities. The direction (*Dir*) is an entity interacting with the agent. The two types of directions, namely source (*So*) and destination (*Dest*) identify the initial and final location of objects to be communicated with, respectively. For example, consider the two scenarios given below:

‘*The bank customer withdraws cash from the ATM_(So)*’,

‘*The ATM reports cash transactions to the Bank_(Dest)*’

In the first scenario, the source of cash is the ATM, and in the second, the Bank is the destination of the cash transactions.

3.2.2 Scenario Authoring Rules

Based on the structure of scenario authoring model, we propose the following scenario-authoring rules. We have developed a domain analysis approach to identify common authoring patterns and their constraints. The formalization of these patterns results in the current set of authoring rules.

In this section, each rule is introduced using the following template <Definition, Comment, Example>. The *definition* explains the contents of the rule. The *comment* is expressed as items to be considered when applying the rule. The *example* component shows a representative example (we show an example from the ATM domain).

Scenario authoring rule 1 (S1)

Definition:

All scenarios should be authored using the following format:

‘*Subject:Agent + Verb + Target:Object + Direction:(Source, Destination)*’

Comment:

The expected scenario prose is a description of a single course of action. This course of action should be an illustration of fulfillment of your goal. You should describe the course of actions you expect, not the actions that are not expected, impossible, and not relevant with regard to the problem domain.

Example:

(The customer)_{Agent} (deposits)_{Verb} (cash)_{Obj} (to the ATM)_{Dest}

Scenario authoring rule 2 (S2)

Definition:

‘*Subject*’ should be filled with an Agent.

Comment:

The agent has the responsibility to fulfill a goal and it may be a human or machine, for example, the designed system itself, an object, or a user.

Example:

(ATM)_{Agent} sends a prompt for code to the user

Scenario authoring rule 3 (S3)

Definition:

‘*Verb*’ should include the properties stated at requirements levels.

Comment:

‘Verb’ should express either the service at service level or the interaction at the interaction level as a transitive verb.

Example:

The customer (withdraws)_{verb} cash from the ATM (service scenario)

The ATM (displays)_{verb} a prompt for amount to the user (interaction scenario)

Scenario authoring rule 4 (S4)***Definition:***

‘Target’ should be an object.

Comment:

The object is a conceptual or physical entity. It can be changed by a ‘Verb’. The change to an object may happen with one or more of its properties such as ownership, place, status, and amount.

Example:

The ATM delivers (the cash)_{obj} to the user

Scenario authoring rule 5 (S5)***Definition:***

‘Direction’ should be either source or destination

Comment:

The two types of *directions*, namely *source* and *destination* identify the origin and destination *objects* for communication. The *source* is the starting point (*object*) of the communication and the *destination* is the ending point of the communication (*object*). Sometimes, the *source* has preposition such as ‘from’, ‘in’, ‘out of’, etc. The *destination* has preposition such as ‘to’, ‘toward’, ‘into’, etc.

Example:

The bank customer withdraws cash (from the ATM)_{so}

Scenario authoring rule 6 (S6)***Definition:***

The designed system and the other agents are used exclusively in instantiating the *Subject* and *Direction* constructs.

Comment:

If the system is used to fill the *subject* slot, the other agents such as human, machine, or an external system should be used to fill the *direction* slot. Thus, the other agents interacting with the system should be treated as candidate actors.

Example:

(The bank customer)_{Agent} withdraws cash (from the ATM)_{so}

Table 1 and figure 3 show goals and scenarios created for the various requirements levels and the scenario authoring model for the ATM example. A scenario consists of actions and states and the flow of actions shows the system’s behavior. The states show necessary conditions for the actions to be fired. In general, a state can become a goal at the lower level (i.e., internal level), and the scenarios corresponding to that goal describe the state in more detail. For example, the state ‘If the card is

'valid' becomes the internal goal 'Check the validity of Card', with corresponding scenarios.

In Figure 3, goals are represented by solid rectangles and the arrows show the relationship among goals. The arrow with dotted line shows the refinement relationship, i.e., the child goals achieve the same parent goal. The bidirectional arrows with solid line connect the sub-goals that "co-achieve the parent goal". For example, there is 'co-achieve' relationship between g1.1 and g1.2, as they achieve 'g1' together.

Table 1. ATM example of rule S1 ~ S6

Goals	Scenarios
Cash withdrawal (g1)	1. The customer withdraws cash from the ATM 2. The ATM reports cash transactions to the bank
Withdraw cash from ATM (g1.1)	1. The ATM receives a card from user <i>If the card is valid, then (state)</i> 2. ATM sends a prompt for code to the user 3. The ATM receives the code from user <i>If the code is valid, then (state)</i> 4. The ATM displays a prompt for amount to the user 5. The ATM receives the amount from user <i>If the amount is valid, then (state)</i> 6. The ATM ejects the card to the user <i>If the user asked the ATM to supply a receipt, then (state)</i> 7. The ATM ejects the printed receipt to the user 8. The ATM delivers the cash to the user

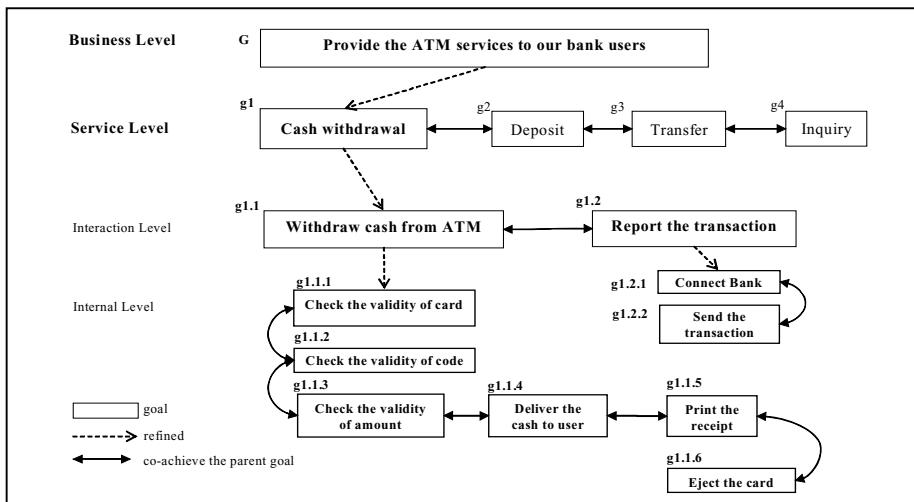


Fig. 3. Partial goal hierarchy with the abstraction levels for ATM example

4 Use Case Conversion

In Section 3.2, a scenario authoring model was proposed to improve the elicitation of the requirements through goals. This section describes how the elicited requirements are used to model use cases. It also shows a way to overcome the lack of elicitation support within UCDA. For use case identification, we propose a relationship model in conjunction with use case conversion rules from goals and scenarios.

The core idea is that the goals and scenarios at the interaction level are used to help construct use cases. A goal at the interaction level is achieved by scenarios and a use case contains the set of possible scenarios to achieve that goal. This is due to the fact that, in our approach, the interaction level focuses on the interactions between the system and its agents. The purpose of the use case specification is to describe how the agent can interact with the system to get the service and achieve his/her goal. Therefore, we propose the following relationship diagram (shown in figure 4) that captures the association between the agents, goals, scenarios and use cases.

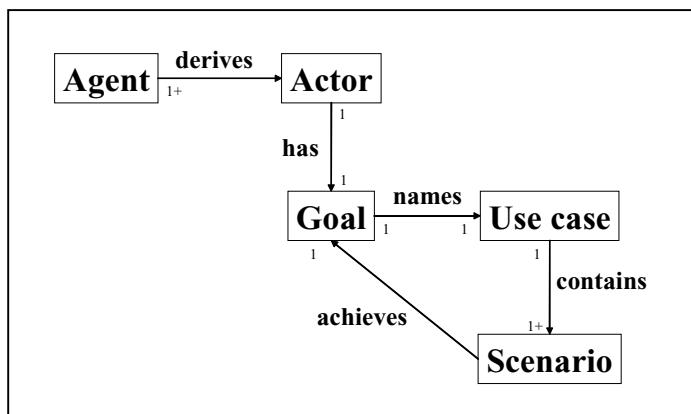


Fig. 4. Relationships between goal and other artifacts

After analyzing each use case, it is augmented with internal activity elements of software-to-be. In our approach, goal and scenario at the internal level represent the internal behavior of the system under consideration to achieve the goals of the interaction level. Accordingly, use cases can be performed by achieving goals at the internal level, and completed by integration of scenarios at that level.

We also propose several guiding rules for the use case conversion using the same template as scenario authoring rules. We restrict the example to the service level goal (e.g: g1 of the ATM example).

Conversion guiding rule 1 (C1)

Definition:

Goals listed at the interaction level become use cases in accordance with figure 4.

Comment:

As mentioned above, goals at interaction level are mapped to use cases. The use cases are named after the goals they correspond to.

Example:

For the goal, g1, goals at the interaction level for ATM example are as follows:

'Withdraw cash from ATM',

'Report the transaction'

These goals become use cases with appropriate descriptions.

Conversion guiding rule 2 (C2)**Definition:**

Agents included in scenarios within a goal and wanting to achieve a goal become primary actors.

Comment:

A goal is achieved by scenarios and several agents may be found in scenarios. As discussed in scenario authoring rules, agents are used to instantiate either *subject* or *direction* objects. Therefore, agents in *subject* or *direction* are all treated as actors except the system that is currently being designed.

Example:

Figure 5 shows the actors that are found in scenarios within goal g1.1 (withdraw cash from ATM) of the ATM example. Sc1.1 has eight actions. All agents corresponding to the *direction* in all the actions are described as 'user'. Thus, in case of Sc1.1, 'user' becomes an actor.

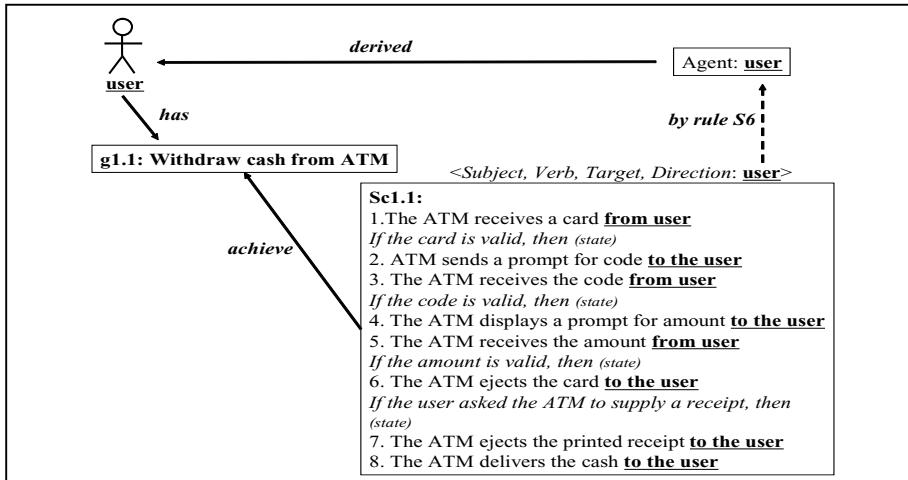


Fig. 5. ATM example of finding actors through scenario structure

Conversion guiding rule 3 (C3)**Definition:**

The States contained in scenarios at the Interaction Level are mapped to internal goals at the Internal Level

Comment:

The state represents necessary conditions for actions to be fired. They can be represented as internal goals at the internal level. Thus, when specifying a use case, we can find more details about states in a goal and scenario at the internal level.

Example:

The state, '*If the card is valid*', of Sc1.1 is described in the sub goal (g1.1.1), 'Check the validity of card' at internal level. The scenarios corresponding to g1.1.1 will also show more details about the state '*If the card is valid*'.

Conversion guiding rule 4 (C4)**Definition:**

If goals at internal level have more than two parent goals, they become another use case with the <include> relationship.

Comment:

Because *include* relationship helps us identify commonality, a goal at internal level with more than two parent goals becomes a use case with <include> relationship.

Example:

In figure 3, the goal, '*Check the validity of card*', has two parent goals, namely, 'Withdraw cash from ATM' and 'Deposit cash into ATM'. Therefore, the goal, '*Check the validity of card*', can become a use case with <include> relationship.

Figure 6 shows the use case diagram for ATM application generated by applying the conversion rules C1through C4. It contains two actors, namely, *User* and *Bank* and six use cases. The use cases associated with the user are *Withdraw*, *Deposit*, *Transfer* and *Inquiry*. These use cases include the "Check Validity" use case, which is associated with the bank along with the *Report* use case.

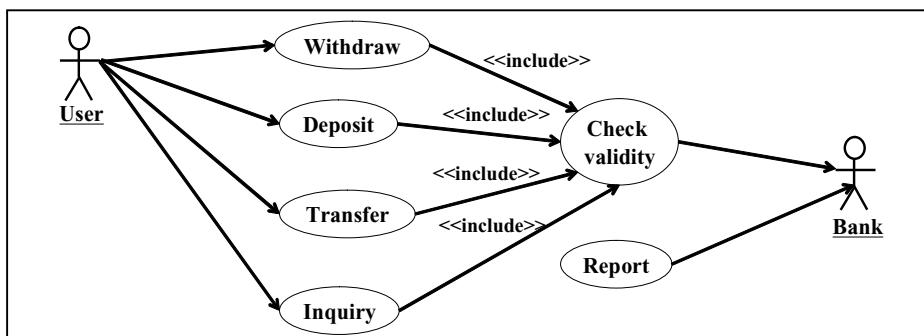


Fig. 6. Use Case Diagram for the ATM Example

5 Conclusion

Our proposed approach overcomes the lack of support for the elicitation process in UCDA and the underlying rationale. It builds on the following two principles: a) both goal modeling and scenario authoring help to cope with the elicitation process in UCDA, and b) use case conversion guidelines help us generate use case diagrams from the output of goal and scenario modeling. There have been some approaches to

create use case diagrams from requirements using some natural language processing techniques [7]. However, due to the varying degrees of details in the requirements text, in many cases, the resulting use case diagrams were found to be practically unusable. In contrast, our proposed approach produces goal descriptions with the same level of specificity, and can be used as adequate input for deriving use case diagrams (potentially automated using a diagramming tool). This promotes customer communication in defining system requirements. Our future work includes further refinement of the scenario structure as well as the authoring and conversion rules. A proof of concept prototype is currently under development and an empirical validation of the approach is planned for the near future.

References

1. B. Regnell, K. Kimbler, A. Wesslén, "Improving the Use Case Driven Approach to Requirements Engineering", RE'95: Second IEEE International Symposium on Requirements Engineering, 1995, pp 1-8.
2. C. Potts, "Fitness for Use: the System Quality that Matters Most," Proc. Third Int'l Workshop Requirements Eng.: Foundations of Software Quality REFSQ '97, pp. 15-28, Barcelona, June 1997.
3. P. Loucopoulos, "The F3 (From Fuzzy to Formal) View on Requirements Engineering," *Ingénierie des Systèmes d'Information*, vol. 2, no. 6, pp. 639-655, 1994.
4. Annie I. Anton, "Goal-Based Requirements Analysis", 2nd International Conference on Requirements Engineering, Colorado, April 15 - 18, 1996, pp. 136-144.
5. C. Potts, K. Takahashi, and A.I. Anton, "Inquiry-Based Requirements Analysis," IEEE Software, vol. 11, no. 2, pp. 21-32, 1994.
6. A. Cockburn, Writing Effective Use Cases, Addison Wesley, 2001.
7. Jolita Ralyté, et al., "Method Enhancement with Scenario Based Techniques", Proceedings of CAiSE 99, 11th Conference on Advanced Information Systems Engineering, 1999.
8. C. Rolland, et al., "Guiding Goal Modeling Using Scenarios", IEEE Transaction on Software Engineering, Vol. 24, No. 12, December 1998, pp 1055-1071.
9. C. Rolland, C. Ben Achour, C. Cauvet, J. Ralyté, A. Sutcliffe, N.A.M. Maiden, M. Jarke, P. Haumer, K. Pohl, Dubois, and P. Heymans, "A Proposal for a Scenario Classification Framework," Requirements Eng. J., vol. 3, no. 1, pp. 23-47, 1998.
10. S. Nurcan, G. Grosz, and C. Souveyet, "Describing Business Processes with a Use Case Driven Approach," Proc. 10th Int'l Conf. CaiSE '98, Lecture Notes in Computer Science 1413, B. Pernici and C. Thanos, eds., Pisa Italy, Springer, June 1998.
11. J.W.Kim, J.T.Kim, S.Y.Park, V.J.Sugumaran, "A Multi view approach for Requirements analysis using goal and scenario", Industrial Management and Data Systems, to be published in 2004.
12. A. Dardenne, S. Fickas and A. van Lamsweerde, "Goal-Directed Concept Acquisition in Requirements Elicitation," Proc. IWSSD-6—Sixth Int'l Workshop Software Specification and Design, pp. 14-21, Como, Italy, 1991.
13. I. Sommerville and P. Sawyer, Requirements Engineering: A Good Practice Guide. Wiley, 1997.
14. V. Plihon, J. Ralyté, A. Benjamen, N.A.M. Maiden, A. Sutcliffe, E.Dubois, and P. Heymans, "A Reuse-Oriented Approach for the Construction of Scenario Based Methods," Proc. Int'l Software Process Assoc. Fifth Int'l Conf. Software Process (ICSP '98), Chicago, pp. 14-17, June 1998.
15. C. Rolland, C. Ben Achour, "Guiding the Construction of Textual Use Case Specifications", Data & Knowledge Engineering Journal, Vol. 25, No. 1-2, 1998, pp. 125-160.

Effectiveness of Index Expressions

F.A. Grootjen and T.P. van der Weide

University of Nijmegen, Faculty of Science, Mathematics and Computing Science,
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands

Abstract. The quest for improving retrieval performance has led to the deployment of larger syntactical units than just plain words. This article presents a retrieval experiment that compares the effectiveness of two unsupervised language models which generate terms that exceed the word boundary. In particular, this article tries to show that index expressions provide, beside their navigational properties, a good way to capture the semantics of inter-word relations and by doing so, form an adequate base for information retrieval applications.

1 Introduction

The success of single-word content descriptors in document retrieval systems is both astonishing and comprehensible. Single-word descriptors are expressive, have a concise meaning and are easy to find¹. This explains the success of word based retrieval systems. Even nowadays, modern internet search engines like Google use complicated ranking systems and provide boolean query formulation, yet are in principle still word based.

The employment of larger syntactical units than merely words for Information Retrieval purposes started in the late sixties [1], but still do not seem to yield the expected success. There are several non-trivial problems which need to be solved in order to effectively make use of multi-word descriptors:

- the introduction of multi-word descriptors boosts precision, but hurts recall.
- the manner of weighting is not obvious, especially in comparison to single-word descriptors which react suitably to standard statistically motivated weighting schemes (such as term frequency/inverse document frequency).
- it is not easy to find distant, semantically related, multi-word descriptors.

The great success of the present statistical techniques combined with such “shallow linguistic techniques” [2] has compelled the idea that deep linguistics is not worth the effort. However, advancements in natural language processing, and the ability to automatically detect related words [3,4] justifies reevaluation.

This article attempts to compare the effectiveness of several language models capable of the unsupervised generation of multi-word descriptors. A comparison is made between standard single-word retrieval results, word n-grams and index expressions.

¹ This might be true for the English language, but for some Asian languages (for example Chinese and Vietnamese) the picture is less clear

2 Method

2.1 Measuring Retrieval Performance

To compare the linguistic models we use standard precision figures measured on 11 different recall values ranging from 0.0 to 1.0, and on the 3 recall values 0.2, 0.5 and 0.8. Subsequently these values are averaged over all queries.

SMART and BRIGHT. The SMART system, developed by Salton [5], played a significant role in experimental Information Retrieval research. This vector space based tool offers the capability to measure and compare the effect of various weighting schemes and elementary linguistic techniques, such as stopword removal and stemming.

It became apparent that extending SMART to the specific needs of modern Information Retrieval research would be rather challenging. The lack of documentation and the style of coding complicates the extension of the system in non-trivial ways. These arguments invoked the decision to redesign this valuable system, preserving its semantic behavior, but written using modern extendible object oriented methods. The resulting system, BRIGHT, has been used in the retrieval experiments presented in this article.

Inside BRIGHT. In contrast to SMART, the BRIGHT system consists of two distinguishable components: the collection specific parser and the retrieval engine. The communication between the constituents is realized by an intermediate statistical collection representation. SMART's capability to specify the input structure, and thus parameterizing the global parser, has been eliminated. Though resulting in the construction of a parser for each new collection², it provides the flexibility of testing elaborated linguistic techniques.

The architecture of BRIGHT is shown in Figure 1.

Test collections. The principal test collection used in this article is the Cranfield test collection [1], a small standard collection of 1398 technical scientific abstracts³. The collection is accompanied by a rather large set of 225 queries along with human assessed relevance judgments. It consists of approximately 14,000 lines of text, and contains almost 250,000 words of which 15,000 unique.

To show that the approach presented is feasible, we tested our findings on the Associated Press newswire collection, part of the TREC dataset. This collection is approximately 800Mb big, containing 250,000 documents and 50 queries. It consists of more than 100,000,000 words of which 300,000 unique.

² Thanks to the object oriented structure of existing BRIGHT parsers, a parser rewrite is relatively easy.

³ The abstracts are numbered 1 to 1400. Abstracts 471 and 995 are missing.

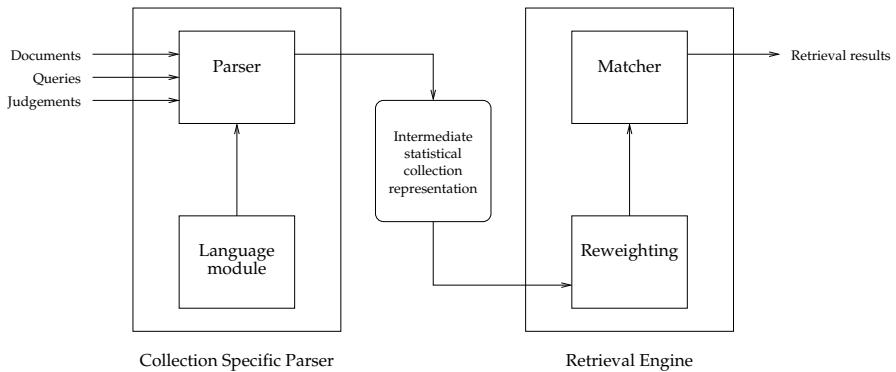


Fig. 1. The BRIGHT architecture

Baseline. The retrieval results of the distinct models will be compared to the standard multiset model, without the use of a special weighting scheme (simply cosine normalization). This baseline will be referred to as **nnc** equivalent to SMART's notation for this particular weighting. The justification for not using more elaborated weighting methods is twofold:

- statistically motivated weighting schemes may mask the linguistic issues
- the purpose of the experiment is to compare different models, not to maximize (tune) retrieval results

Although one of the language models outperforms term frequency/inverse document frequency weighting (**atc**), this is of less importance regarding the scope of this article.

2.2 Beyond the Word Boundary

A key issue in Information Retrieval is to find an efficient and effective mechanism to automatically derive a document representation that describes its contents. The most successful approach thus far is to employ statistics of individual words, ignoring all of the structure within the document (the multiset model). Obviously indexing is not necessarily limited to words. The use of larger (syntactical) units has been the subject of research for many years. The benefit is clear: larger units allow more detailed (specific) indexes and are a way to raise precision. On the other hand, the rare occurrences of these units will hurt recall. We describe two indexing models that exceed the word boundary, namely *word n-grams* and *index expressions* and compare their retrieval performance using BRIGHT

Word n-grams. The word n-gram model tries to capture inter-word relations by simply denoting the words as ordered pairs, triples etc. In effect, the n-gram model extends the multiset model with sequences of (at most n) consecutive

words in the order which they appear in the text. Consider the following document excerpt:

An experimental study of a wing in a propeller slipstream was made in order to determine the spanwise distribution of the lift increase due to slipstream at different angles of attack of the wing and at different free stream to slipstream velocity ratios.

The 2-gram model will add, besides each word individually, the descriptor '*propeller slipstream*' which is obviously meaningful. The model is rather imprecise however, since adding the descriptor '*and at*' will probably not contribute to retrieval performance. Some researchers therefore only add n-grams consisting of non-stopwords, or consider an n-gram only worthwhile if it has a (fixed) frequency in the collection.

Index expressions. As already shown before, simply using sequences of words for indexing purposes has some drawbacks:

- Sequential words are not necessarily semantically related.
- Sometimes words are semantically related, but are not sequential.

It seems plausible to look for combinations of words that are semantically related. In [3] an algorithm is presented which is capable of finding relations between words in natural language text. These relations form a hierarchical structure that is represented by index expressions.

Index expressions extend term phrases which model the relationships between terms. In this light, index expressions can be seen as an approximation of the rich concept of noun (and verb) phrases. Their philosophical basis stems from Farradane's *relational indexing* [6,7]. Farradane projected the idea that a considerable amount of the meaning in information objects is denoted in the relationships between the terms.

Language of index expressions. Let T be a set of terms and C a set of connectors. The language of index expressions is defined over the alphabet $\Sigma = T \cup C \cup \{(),\}$ using structural induction:

- (i) t is an index expression (for $t \in T$).
- (ii) $e_1 \circ c(e_2)$ is an index expression (for index expressions e_1, e_2 and $c \in C$).

In this definition, the \circ operator denotes string concatenation. If there are no means for confusion, we omit the parentheses when writing down index expressions.

The structural properties of these expressions provide special opportunities to support a searcher in formulating their information need in terms of a (information system dependent) query. The resulting mechanism is called *Query by Navigation* [8]. In [9] this mechanism is described from a semantical point of view. By employing the relation between terms and documents, concepts are

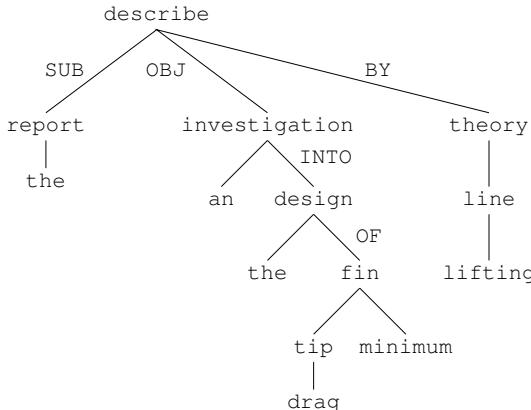


Fig. 2. Tree structure of example sentence.

derived which are used as navigational pivots during Query by Navigation. Index expressions have been motivated and validated of their potential to support interactive query formulation, without assuming that the searcher is familiar with the collection. The rationale of the approach is that a searcher may not be able to formulate the information need, but is well capable of recognizing the relevance of a formulation.

Consider the following input sentence:

The report describes an investigation into the design of minimum drag tip fins by lifting line theory.

The corresponding parsed index expression is:

```
describe SUB (report IS (the)) OBJ (investigation IS (an) INTO
(design IS (the) OF (fin IS (tip IS (drag)) IS (minimum)))) BY
(theory IS (line IS (lifting)))
```

whose structure is visualized in figure 2. Note that in this index expression, the verb-subject and verb-object relations are represented by the `SUB` and `OBJ` connectors, while apposition is represented by the `IS` connector. Using this index expression it is possible to generate subexpressions. Simply put, subexpressions of an index expression are like subtrees of the tree structure. Preceding a more formal definition, we will introduce power index expressions, a notion similar to power sets.

Power index expressions. Let $e = t \circ_{i=1}^k c_i e_i$ be an index expression. The set $\Lambda(e)$ of *lead expressions* belonging to e is defined as follows:

$$\Lambda(e) \stackrel{\text{def}}{=} \bigcup_{(b_1, \dots, b_k) \in \{0,1\}^k} t \circ_{i=1}^k (c_i \Lambda(e_i))^{b_i}$$

The power index expression belonging to e , denoted by $\mathcal{P}(e)$, is the set

$$\mathcal{P}(e) \stackrel{\text{def}}{=} A(e) \cup \bigcup_{i=1}^k \mathcal{P}(e_i)$$

Using this definition we can now formally define what a subexpression is:

Subexpression. Let e_1 and e_2 be two index expressions, then:

$$e_1 \sqsubseteq e_2 \stackrel{\text{def}}{=} e_1 \in \mathcal{P}(e_2)$$

Among the subexpressions in our example sentence we find ‘**describe BY theory**’, clearly non-sequential words having a strong semantic relation.

Instead of using all subexpressions as descriptors, we restrict ourselves to subexpressions having a maximum length.⁴ In this article we evaluate the retrieval performance for 2-index, 3-index and 4-index subexpressions. Note that a similar linguistic approach which creates head-modifier frames [10] is essentially a cutdown version of index expressions, while their unnesting into head-modifier pairs generates index expressions of length 2.

3 Results

3.1 Validation Results

Baseline. The Cranfield baseline experiment yields the following results:

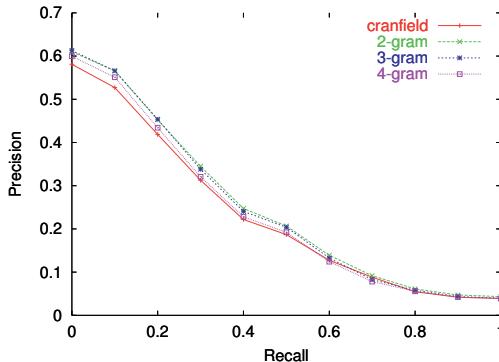
scheme	11-pt average	3-pt average
nnc	0.2363 (100.0%)	0.2201 (100.0%)

Word n-grams. We performed retrieval runs using BRIGHT on n-grams with $1 \leq n \leq 4$ and weighting scheme **nnc**. $n = 1$ produces the multiset model (baseline). Note that, for example, the run with $n = 3$ uses word sequences of length 3 and those smaller as semantical units.

n	units	11-pt average	3-pt average
1	7223	0.2363 (100.0%)	0.2201 (100.0%)
2	79675	0.2554 (108.1%)	0.2401 (109.1%)
3	230870	0.2519 (106.6%)	0.2384 (108.3%)
4	422554	0.2422 (102.5%)	0.2273 (103.3%)

The results of different n-gram runs are depicted in figure 3. It is easy to see that all n-gram runs perform better than the baseline. The best improvement is obtained in the high precision - low recall area, which is not surprising, since n-grams have a more specific meaning, but occur less frequently than words. The best results are obtained for $n = 2$. As anticipated, the retrieval performance decreases slightly when n is increased, because more ‘meaningless’ units are generated than ‘meaningful’ units.

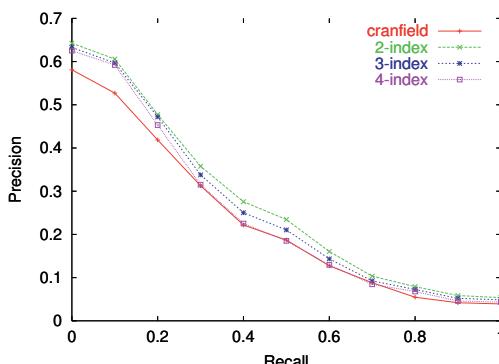
⁴ The length of a index expression is the number of terms that occur in the expression.

**Fig. 3.** n-grams compared

Index expressions. As with n-grams, we use BRIGHT to measure retrieval performance for different maximum lengths of index expressions. Again, for $n = 1$ the multiset model is produced which functions as baseline. The results are presented below and visualized in figure 4.

n	terminals	11-pt average	3-pt average
1	7223	0.2363 (100.0%)	0.2201 (100.0%)
2	68061	0.2771 (117.3%)	0.2635 (119.7%)
3	206034	0.2645 (111.9%)	0.2517 (114.4%)
4	429084	0.2515 (106.4%)	0.2353 (106.9%)

The best results are obtained for $n = 2$. Obviously, long index expressions have high descriptive power, but are rare. So, similar to n-grams we notice the highest improvement in high precision - low recall area. Interesting is that the 4-index

**Fig. 4.** Index expressions

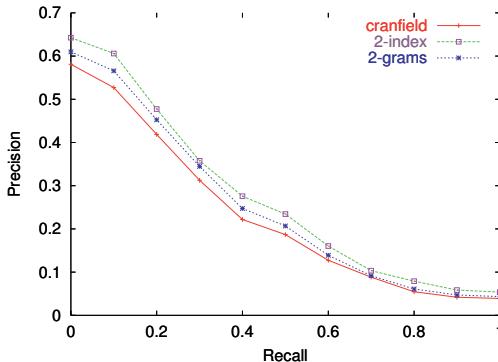


Fig. 5. Index expressions vs. n-grams

starts off relatively good, but as soon precision drops under 0.4 it is almost indistinguishable from the baseline.

Comparing n-grams with index expressions. Combining the results of the previous two sections we are capable of comparing the retrieval performance of index expressions with the performance of n-grams (see figure 5). 2-index outperforms 2-ngrams throughout the recall spectrum. The gain in performance achieved by 2-ngram is doubled by 2-index. This stresses the semantical validity of automatically generated index expressions.

3.2 TREC Results

We performed two retrieval runs on the Associated Press collection: a standard word based retrieval run (baseline) and the 2-index run.

type	11-pt average	3-pt average
word	0.0272 (100.0%)	0.0142 (100.0%)
2-index	0.0620 (227.9%)	0.0380 (267.6%)

The relatively low score for the baseline is primarily due to the absence of an elaborated weighting scheme. Nevertheless, the 2-index run (with the same simple weighting scheme) scores significantly better.

The resulting precision-recall data is shown in see figure 6.

3.3 Weighting Index Expressions

In the previous experiments we treated index expressions in the same manner as terms. Because index expressions often consist of more than one word it seems reasonable to give them a higher weight than simple (single word) terms.

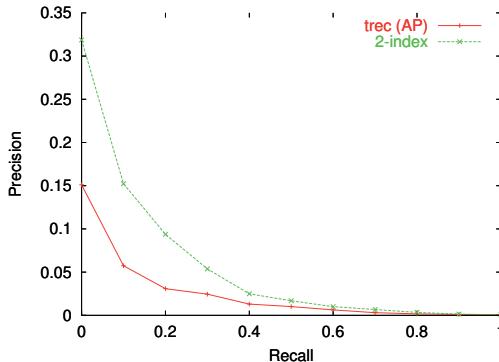


Fig. 6. Associated Press words vs. 2-index

The following experiment compares the 11-pt average retrieval performance of index expressions for several weight factors. In figure 7 we show how the retrieval performance is effected by adjusting the weight factor of index expressions having length 2. The best retrieval performance is obtained using a weight factor of approximately 2. The minimal improvement of 5% for weight factor 0 might seem strange at first glance; one might expect a gain of 0%, since eliminating index expressions with length 2 leaves us with plain terms. However, there is a mild form of stemming in the index expression model which contributes to this small gain in retrieval performance.

Studying the retrieval results of index expressions with length smaller or equal to 3, there are two changeable parameters; the weightfactor of index expressions of length 2, and the weightfactor of index expressions of length 3. This results in the 3d plot depicted in figure 8. Again, the maximal performance is

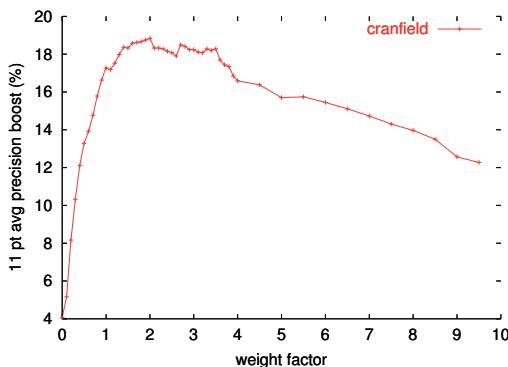


Fig. 7. Influence of weight factor

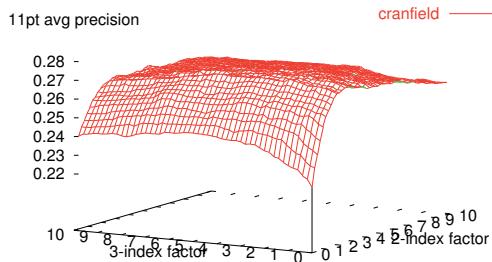


Fig. 8. Influence of weight factors

obtained by doubling the weight of index expressions with length 2. For index expressions with length 3 the picture is vague. Apparently the weightfactor (and the importance of these index expressions) is less obvious. According to the data, the maximal combined performance is for (2.3,3.1).

4 Conclusions

As shown in this article, index expressions are suitable for capturing the semantics of inter-word relations. Experiments show that 2-indexes perform better than standard word-based retrieval runs, especially on the large TREC collection where the retrieval performance is more than doubled.

Compared to 2-grams, index expressions show an improvement of 10% on the small Canfield collection. Due to the enormous number of possible 2-grams in large collections, it was unfeasible to compare 2-grams and 2-indexes for the TREC collection.

In situations where the structure of index expressions can be exploited (as in query by navigation) they seem to form a beneficial alternative to term based systems, which is validated in [11].

References

1. Cleverdon, C.: The cranfield tests on index language devices. *Aslib Proceedings* (1967) 173–194
2. Sparck Jones, K.: Information retrieval: how far will *really* simple methods take you? In Hiemstra, D., de Jong, F., Netter, K., eds.: *Proceedings Twente Workshop on Language Technology 14.* (1998) 71–78
3. Grootjen, F.: Indexing using a grammarless parser. In: *Proceedings of the 2001 IEEE International Conference on Systems, Man and Cybernetics, (NLPKE 2001), Tucson, Arizona, USA* (2001)

4. Kamphuis, V., Sarbo, J.J.: Natural Language Concept Analysis. In Powers, D.M.W., ed.: Proceedings of NeMLaP3/CoNLL98: International Conference on New Methods in Language Processing and Computational Natural Language Learning, ACL (1998) 205–214
5. Salton, G., ed.: The SMART retrieval system. Prentice Hall, Englewood Cliffs (1971)
6. Farradane, J.: Relational indexing part i. *Journal of Information Science* **1** (1980) 267–276
7. Farradane, J.: Relational indexing part ii. *Journal of Information Science* **1** (1980) 313–324
8. Bruza, P., van der Weide, Th.P.: Stratified hypermedia structures for information disclosure. *The Computer Journal* **35** (1992) 208–220
9. Grootjen, F., van der Weide, Th.P.: Conceptual relevance feedback. In: Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics, (NLPKE 2002), Tunis (2002)
10. Koster, K.: Head/modifier frames for information retrieval. Technical report, PEKING project (2003)
11. Grootjen, F., van der Weide, Th.P.: Conceptual query expansion. Technical Report NIII-R0406, University of Nijmegen (2004)

Concept Similarity Measures the Understanding Between Two Agents

Jesus M. Olivares-Ceja and Adolfo Guzman-Arenas

Centro de Investigacion en Computacion (CIC), Instituto Politecnico Nacional, Mexico
a.guzman@acm.org

Abstract. When knowledge in each agent is represented by an ontology of concepts and relations, concept communication can not be fulfilled through exchanging concepts (ontology nodes). Instead, agents try to communicate with each other through a common language, which is often ambiguous (such as a natural language), to share knowledge. This ambiguous language, and the different concepts they master, give rise to *imperfect* understanding among them: How well concepts in ontology O_A map¹ to which of O_B ? Using a method *sim* that finds the *most similar concept* in O_B corresponding to another concept in O_A , we present two algorithms, one to measure the similarity between both concepts; another to gauge **du**, the *degree of understanding* that agent A has about B's ontology. The procedures use word comparison, since no agent can measure **du** directly. Method *sim* is also compared with *conf*, a method that finds the *confusion* among words in a hierarchy. Examples follow.

1 Introduction and Objectives

The easiest thing for two agents seeking communication (information exchange) is agreeing first in *what to communicate*, how and in what order, and then, doing it. Unfortunately, this requires a “wiser creature” (a programmer, or a Standards Committee) to establish these agreements.² In this paper we will assume that no creature of this kind is to be used. Then, what can an agent do to meaningfully communicate³ with other agents (or persons), even when it/they had not made any very specific commitment to share a private ontology and communication protocol? Concept communication can not be fulfilled through direct exchange of concepts belonging to an ontology, since *they do not share* the same ontology, and O_A and O_B are in different *address spaces*. Instead, they should use a common language for communication. Lucky agents can agree on a language whose words have a *unique meaning*. Others need to use an ambiguous language (such as a natural language) to share knowledge. This gives rise to imperfect understanding and confusion.

In a different approach, [16] proposes to use natural language *words* as concepts.

¹ O_A and O_B are the ontologies of agents A and B, in the rest of this document.

² That is, the agents need to communicate *in order to agree* about how to communicate.

³ Agent A communicates with B “in a meaningful way” when A moves towards its goals as the information exchange progresses. Each could be a person or a piece of software.

When one agent is talking to another, can the talker discover on what it (or the listener) is confounded? Is it possible to measure this mix-up? How can I be sure you *understand* me? Can I *measure* how much you understand me? Can you *measure* it? These questions have intrigued sociologists; they are also relevant to agents “not previously known to each other”⁴ trying to “interact with free-will”,⁵ for which they have to exchange knowledge. The paper gives answers for them.

Knowledge is stored in concepts (Cf. §1.2), which are mapped by the talker into words of the communication language; perceived words are internalized as concepts by the listener. If the concepts exchanged are animals and plants, Latin is fine: *Felix Leo* represents the concept *lion-león-loin*⁶ while *Cannabis Indica* stands for the concept *marijuana*. Other examples of words or symbols with a unique (universal) meaning: 4, π , Abelian group, Mexico, (23°22'57"N, 100°30'W), Abraham Lincoln, Berlin Symphony Orchestra. There are also semi-universal (popular) conventions [such as standard naming for chemical compounds, the Catalog for books of the Library of Congress, or the USA Social Security Number], which provide non-ambiguity for those who adhere. If two agents can select a non-ambiguous language (each of its words maps exactly to one concept) or convention to exchange concepts, great. Otherwise, they have to settle for an ambiguous language, such as English [7].

If two agents do not share a concept (figures 1 and 2), at least partially, they can not communicate it or about it. Thus, a measure of the amount of understanding can be the number of concepts they share, and *how well* they share them.⁷ We will sharpen these measures for both cases: the ambiguous and the non ambiguous communication language.

⁴ It is much easier to design the interaction (hence, the exchange of concepts) between two agents (each could be a person, or a piece of software), when the *same* designer designs both agents. In this sense, “they previously know each other:” each agent knows what the other expects, when, and the proper vocabulary to use. In contrast, our approach will allow *my* agents to interact with *yours*, as well as with Nigerian agents.

⁵ Not-free will or canned interactions are those that follow defined paths. For instance, the interaction between a program and a subroutine it calls, where the calling sequence (of arguments) is known to both. Free will requires goals, resources, and planning [16].

⁶ We represent concepts in Courier font. A concept is language-independent: the concept *cat* is the same as the concepts *gato-gata*, *gatto-gatta*, *chat-chatt*, *Katze*, *KOT-KOIIKA*, meaning “a small domestic feline animal.” Concepts appear in English in this paper, for readers’ benefit.

⁷ Knowledge is also stored in the relations (or verbs, actions, processes) between objects (or nouns, subjects): A balloon can explode. It is also stored in the properties (or adjectives, adverbs) of these nouns and relations. The value of a property also contains information. A relation such as *explode* can also have relations (such as *father_of*, *speed_of_explosion*) which can also be nodes (concepts) in the ontology. Words are *not* nodes, but they are attached to the node they *denote*. The definition of ontology in §1.2 brings precision to these ideas: *a concept is always a node of an ontology* (and vice versa), whereas a relation and the other object related or linked (relations are arcs among two objects) may be a concept (node) or just a word or a token as defined in footnote 10. A relation that is just a token is called a property. If the “other object” is just a token, it is called a value. Examples: (*cat* *drinks* *milk*), (*balloon* *price* *50_cents*), (*balloon* *inflated_by* *JFKennedy*), (*balloon* *vanished* *languidly*).

1.1 Related Work

An ancestor of our *sim* (§3.1) matching mechanism is [3], based on the theory of analogy. Most work on ontologies involve the construction of a single ontology (for instance, [13]), even those that do collaborative design [10]. Often, ontologies are built for man-machine interaction [12] and not for machine-machine interaction. Work [2] identifies conceptually similar *documents* using a single ontology. Sidorov [4] does the same using a topic hierarchy: a kind of ontology. Linguists [14] identify related *words* (semantic relatedness), not *concepts*, often by statistical comparisons.

Huhns [11] seeks to communicate agents sharing a single ontology, such as O_D (Fig. 1). The authors are motivated [7] by the need of agents to communicate with unknown agents, so that not much *a priori* agreement between them is possible.⁴

```

thing (thing, something, object) {
    living (organism, life, being, creature, living thing) {
        animal (animal) {
            man (man, person, human being, woman, girl)
            [eats = apple, peach ] }

        plant (plant, vegetal) }

    inanimate (inanimate object, tangible object) {
        rock (rock, stone)

        food (food, foodstuff, provisions) {
            solid_food (solid food){
                apple (apple)[shape = round][color = red, yellow, green]
                peach (peach) [color = orange, green, yellow]
                [shape = round]
                bread (bread) [color = brown] }

            liquid_food (liquid, drink) {
                water (water)
                coffee (coffee, coffee drink, espresso) [color = black]
                milk (milk) [color = white]
                beer (beer) wine(wine) } } }

        abstract_thing (intangible object, abstract object, abstract thing) }
    
```

Ontology
 O_D

Fig. 1. Ontology O_D : Foods. Concepts appear in courier font, words denoting a concept are in parentheses. Properties are inside [], such as [color = red, yellow, green]

Simple measurements [9] between *qualitative values* (“words”) belonging to a *hierarchy*, find out how close two values are: the *confusion* between these values is measured. More at §3.1. Also, there is much work on tree distances.

With respect to the communication language, we prefer, in decreasing order:

1. A language whose tokens (words, atoms) are formed by concepts [8];
2. One with unambiguous tokens (a token represents only one concept). Examples: the Natural Numbers; the Proper Nouns;
3. One where each token has a small number of ambiguities, for instance, a natural language [14];
4. One where each token points to a foreign address space, thus representing a black box that can only be compared with = and ≠. Example: a language consisting of tokens such as “brillig”, “toves,” “borogove,” “mome”, “outgrabe”, “fromelic,” “meratroping”...

Approaches and experimental results through semantic analysis are in book [1].

1.2 Definitions

Level of a node in a tree. The root has level 0. A node has level $\lambda + 1$ if his father has level λ .♦⁸

Concept. An object, relation, property, action, process, idea, entity or thing that has a name: a word(s) in a natural language. ♦ Examples: peak-uttermost, angry-mad, to_fly_in_air. So, *concepts have names*: those words (or word phrases, such as New York City) used to denote them. A concept is unambiguous, by definition.⁶ Unfortunately, the names given by different people to concepts differ and, more unluckily, the same word is given to two concepts (examples: words peak; fly; mad). Thus, *words are ambiguous, while concepts are not*. A person or agent, when receiving words from a speaker, has to solve their ambiguity in order to understand the speaker, by mapping the words to the “right” concept in his/her/its own ontology. The mapping of words to concepts is called *disambiguation*.

There are also composite or complex concepts, such as “to ski in a gently slope under a fair breeze while holding in the left hand a can of beer.” These can be shared with other agents, too, but they do not possess a name: they have not been reified.

Ontology. It is a formal explicit specification of a shared conceptualization [5].♦ It is a taxonomy of the concepts we know.⁹ We represent an ontology as a graph where each node is a concept and the arcs are relations to other concepts or tokens.¹⁰ Some relations are concepts (such as subset, member_of, part_of, eats-ingests, lives_in); others are just tokens (which are called properties), represented in Times font, such as “color.” Each relation links a node with other node (a concept) or with a token,¹⁰ in the last case the token is called the value of the relation; for instance, “blue.” In addition, associated to each node are the words that represent or denote that concept. The relation subset is represented by { }. §5 suggests a better representation. Examples: O_D and O_T in figures 1 and 2.

Size of O_A . Written as $|O_A|$, is the number of concepts in O_A . ♦

Teaching and learning. Agent T *teaches* agent S, and S *learns* from T, a set of concepts that T knows, if T patiently (incrementally) sends appropriate trios [of the

⁸ Symbol ♦ means: end of definition. Having a name in a shared language means that it is known to many people.

⁹ Each concept that I know and has a name is shared, since it was named by somebody else.

¹⁰ These tokens are words or strings of the types 2, 3, 4 of the list at the end of §1.1.

form (concept relation concept)] to S such that S can build new nodes on its ontology, resembling those nodes already present in T. ♦ Agent T must often query S to see if “it has learned right”, and to resolve contradictions or questions from S arising from its previous knowledge. More at §3.3.

2 Measuring the Amount of Knowledge

How much does an agent know?

The amount of knowledge an agent has, is the number of concepts in its ontology.

- ♦ It is $|O_A|$. It is the area under the histogram of concepts (see Clasitex in [6]). This definition will be revisited in §3. *To know* a given discipline is to possess many concepts from that discipline.♦

How much does an agent know of the knowledge possessed by other agent? By comparing their histograms of concepts, we can find out that A knows twice more concepts than B about Numismatics, and that A and B know the same number of concepts about dinosaurs. A more accurate measure is given in the next section, where it is called the *degree of understanding*.

3 Measuring the Degree of Understanding

Two definitions are needed to quantify the (imperfect) grasp of a concept by an agent. One is *the most similar concept* in O_B to concept c_A in O_A ; the other is *the degree of understanding* of B about the knowledge of A, which A keeps in O_A .

Assume that A knows that a $diprotodon_A^{11}$ is a mammal of the Tertiary Age, but B knows that a $diprotodon_B$ is a bear-like animal, of prehistory, has fur, two long milk teeth, and a size 5 meters long and 2 meters tall. All these concepts can be perfectly represented with the tools of §§1-2, since the trio ($diprotodon$ - $fossil^{12}$ $skin$ - $epidermis$ fur - $hair$) is present (and true) in O_B and absent in O_A , and similarly for the other relations and concepts. Each concept and each trio of O_A is known by agent A, and the same is true for O_B and B. But it is also advantageous to “concentrate on the nouns” and to say that $diprotodon$ is vaguely or less known to A than to B, since $diprotodon_B$ has more properties and more relations in O_B than $diprotodon_A$ in O_A .

¹¹ We use sub index A to stress the fact that $diprotodon_A$ belongs to O_A .

¹² When explaining concepts to the reader (of this paper) through English words, the convention in [15] is good: we use the word for the concept followed by a dash followed by the word that represents the father of the concept. This provokes little ambiguity in the reader, and it has been programmed in *sim*, the mapper of a concept to the closest concept in another ontology (§3.1). Thus, we write *star-person*, *star-animal*, *star-astronomic_body*, *star-adornment*, for the four meanings of word *star*.

```

thing (thing, something, object, entity) {
    physical_object (concrete object, physical object) {
        living_creature (creature, life form, live being, living creature, organism, being) {
            animal (animal) {
                invertebrate (invertebrate) {
                    insect (insect, bug) {
                        fly_animal(fly, flies) cockroach (cockroach) flea(flea) }
                    mollusk (mollusk) }
                vertebrate (vertebrate) {
                    reptile (reptile) {lizard (lizard) iguana (iguana) }
                    batrachians (batrachians) {frog (frog) }
                    mammal (mammal) { zebra (zebra)
                        rodent (rodent) {
                            rat (mouse, mice, rat) {
                                domestic_rat (domestic rat)
                                country_rat (country rat) }
                            mole-rodent (mole) }
                        fox (fox) cat(cat, kitty) dog(dog) lion(lion, cub) donkey(donkey)
                        man (man, men, woman, women, person, people, human being,
                            Homo Sapiens, boy, girl, child, miss, mister, sir)
                        [eats = tropical_plant, citrus] }
                    bird (bird) { chicken (chicken, hen, cock, rooster, chick, poultry)
                        duck (duck) parrot (parrot) hawk (hawk) }
                    fish (fish) } }
                plant-creature (plant, vegetal) {
                    tropical_fruit (tropical fruit) {
                        coconut (coconut) mango (mango, mangoes) }
                    citrus (citrus, citric) { lemon (lemon) orange (orange)
                        tangerine (tangerine) [color = orange]
                        rare_fruit [color = green] [texture = smooth] [size = 5cm] } }
                    bacteria (bacteria, microorganism) }
                artificial_object (artifact, artificial object) }
            abstract_object (imaginary object, abstract thing, abstract concept) }
}

```

Ontology O_T

Fig. 2. Ontology O_T . Properties are of the form [property-name = value], where the property name or the value may be concepts as in [eats = tropical_plant, citrus] or tokens of the types of footnote 10, as in [color = orange].

Definition. The *degree of knowledge* of A about a concept c is a number between 0 and 1, obtained by counting the number of relations containing c in O_A , adding the number of properties of c in O_A , and dividing into the similar calculation for c in the total ontology.¹³ ♦ The closer it is to 1, the less *imperfect* is A's knowledge of c . ♦ This definition is impractical to use since the total ontology is out of our reach. Thus, we shall compute instead the degree of knowledge of an agent *with respect to another agent*, which we refer to in §3.2 as the *degree of understanding* of A about O_B : how much A understands about what B knows; how well each concept of A maps into the

¹³ The ontology of an agent that knows much, if not everything.

corresponding (most similar) concept in B. Our examples refer to figure 2. First, we need to find the concept in O_B most similar to one given (through words describing it) by A, belonging of course to O_A .

3.1 Finding the Concept in O_B Most Similar to a Given Concept in O_A

Algorithm *sim* [8] (called “*hallar(c_A)*” or COM in [15]) finds *the most similar concept c_B in O_B to concept c_A in O_A* . Agent A makes known concept c_A to B by sending to B words¹⁴ denoting c_A , and also sending words denoting c_A ’s father. Also, *sim* returns a similarity value $sv \in [0, 1]$ expressing how similar was c_B to c_A .

If c_B is the concept most similar to c_A , it is not necessarily true that c_A is the concept most similar to c_B . *Function sim is not symmetric.* Example: A physician P knows six kinds of hepatitis, including the popular hepatitis type A, while John only knows hepatitis. Each of the six hepatitis of P finds John’s hepatitis as “the most similar concept John has,” while John’s hepatitis best maps into P’s *type_A_hepatitis*. P knows more than John, so P can select a better target in his rich ontology for John’s vague concept. *John can not make such selection.*

The function *sim* is only defined between a concept c_A in O_A and *the most similar concept c_B in O_B* . Extensions *sim'* and *sim"* appear below.

Who runs sim? Who compares these two concepts, since they belong to different ontologies? That is, who runs *sim*? Either agent A or B can execute it, since *sim* compares words, not concepts. But, when A runs *sim*, it needs the collaboration of B (and vice versa), which has to provide words to be used by *sim* (thus, by A). Also, even when A executes *sim* producing c_B as result, A can not “have” or “see” c_B ; it is a pointer to the memory O_B , a meaningless pointer for A, such as the tokens of point 4 of §1.1. The most of what A can see of c_B is (1) the words which denote c_B , as well as (the words for) the relations of c_B ; (2) corresponding words for the father, grandfather, sons... of c_B (and words for *their* relations); (3) value sv , indicating how similar that elusive c_B is to its (very solid) c_A . In fact, A still has c_A as “the concept I have been thinking all along.” When B runs *sim*, B can see, of course, c_B , but it can not “see” or “grasp” c_A . The most of what B can see of c_A is that “agent A wants to talk about something of which the closest I have is c_B ”.¹⁵ B can sense from the words sent to it by A differences between its solid c_B and the elusive c_A of A. More in §3.3.

¹⁴ By §1, A can not send any *node* of O_A to B. If later in the algorithm, A needs to send a relation of c_A to B (such as *color*), it sends the words (*color*, *hue*) corresponding to such relation *color*. No concepts travel from A to B or vice versa, just words denoting them.

¹⁵ It will not help if A is more cooperative. For instance, dumping all its O_A into B’s memory will not help B, who will still see a tangled mesh of meaningless pointers. Well, not totally meaningless –some understandable words are attached to each node (concept). Yes: B can slowly understand (untangle) O_A by comparing each concept in O_A with every concept in its own O_B –that is, by using *sim*! See §5 “Suggestions for further work.”

Generalizing sim. Function $\text{sim}'(c_A, d_A)$ for two concepts belonging to the *same* ontology, is defined as $1/(1+\text{length of the path going from } c_A \text{ to } d_A \text{ in the } O_A \text{ tree})$. ♦ The path is through subset and superset relations; its length is the number of such arcs traveled. $\text{sim}'(c_A, d_A) \in (0, 1]$. sim' is symmetric. Example: Figure 3.

Relation to confusion. In [9], the confusion $\text{conf}(c_A, d_A)$ occurring by using c_A instead of d_A , is defined as the length of the descending¹⁶ path from c_A to d_A . ♦ This definition holds for hierarchies; it is here extended to ontologies. If we had defined $\text{sim}'(c_A, d_A) = (1/(1 + \text{length of the descending path going from } c_A \text{ to } d_A \text{ in the } O_A \text{ tree}))$, we would have had $\text{sim}'(c_A, d_A) = 1/1 + \text{conf}(c_A, d_A)$. We prefer, for ontologies, the first definition of sim' , since it is symmetric, while conf is not. Example: for ontology O_D of figure 1, $\text{conf}(\text{liquid_food}, \text{food}) = 0$; the confusion when using liquid_food instead of food is 0, since liquid food is food. But $\text{conf}(\text{food}, \text{liquid_food}) = 1$; when I want liquid food but I am given food, there is an error of 1 (a small error, you could think). More in figure 3.

Confusion and similarity for concepts x and y belonging to the same ontology O_D of figure 1	$\text{conf}(x, y)$; confusion in using x instead of y	$\text{conf}(y, x)$; confusion in using y instead of x	$\text{sim}'(x, y) = \text{sim}'(y, x)$; similarity between x and y
$x = \text{man}, y = \text{living}$	0	2	1/3
$x = \text{peach}, y = \text{bread}$	1	1	1/3
$x = \text{bread}, y = \text{water}$	2	2	1/5
$x = \text{water}, y = \text{man}$	3	4	1/8
$x = \text{bread}, y = \text{coffee}$	2	2	1/5
$x = \text{thing}, y = \text{wine}$	4	0	1/5

Fig. 3. Examples of confusion and similarity (sim') for two concepts of the same ontology (O_D , figure 1). Function conf is not symmetric; sim' is.

For similarity between *any* two objects in different ontologies, we have:

$\text{sim}''(c_A, d_B)$ is found by making first $s_1 = sv$ returned by $\text{sim}(c_A)$ [this also finds c_B , the object in O_B *most* similar to c_A]; then, find $s_2 = \text{sim}'(d_B, c_B)$. Finally, $\text{sim}''(c_A, d_B) = s_1 s_2$. ♦

3.2 Degree of Understanding

The value sv found in $c_B = \text{sim}(c_A)$ in §3.1 can be thought of as the degree of understanding that agent B has about concept c_A . A concept c_A that produces $sv=0$ indi-

¹⁶ Going towards more specialized concepts. Using a person from Dallas when I want to use a Texan person, confusion is 0; using a Texan person when a Dallas person is needed causes confusion=1; using a US person causes confusion=2.

cates that B has understanding 0 (no understanding) about that c_A . Averaging these sv 's for all concepts in O_A , gives the degree of understanding that agent B has about the whole ontology O_A of A.¹⁷ It is as if agent A examines and asks B, for each concept $c_A \in O_A$, «Do you understand what is c_A ?» «How much do you understand c_A ?» At the end, A and B have a good idea of the understanding of B (about O_A).

The *degree of understanding* of B about O_A , $du(B, O_A) = \{\text{sum over all } c_A \in O_A \text{ of } sv \text{ returned by } sim(c_A)\} / |O_A|$. ♦ It is the average of the sv 's. Similarly, we can measure the degree of understanding of B about *some region* of O_A . ♦ Function *du* is not symmetric. In general, an agent understands some regions better than others. If $|O_A| \gg |O_B|$, then $du(B, O_A)$ is small: B knows little about O_A , even if all parts of O_A known to B were to have $sv=1$.

$du(B, O_A) \leq 1$; in regions where B knows more than A, $du = 1$. Example: assume agent T has the ontology O_T of figure 2 and agent N has ontology O_N of figure 4. Then, figure 5 shows the concept c_T most similar to each $c_N \in O_N$, as well as the corresponding similarity value sv . Thus, $du(T, O_N) = (\Sigma sv)/|O_N| = 9.58/15 = 0.64$ is the degree of understanding that agent T has about ontology O_N .

```
living_creature (organism, being, living creature) {
    animal (animal)
        frog (frog, tadpole)    iguana (iguana)
        diprotodon (diprotodon) }
    plant-creature (plant, vegetal) {
        big_plant (big plant, large plant) {
            coconut (coconut)    mango (mango) }
        small_plant (small plant) {
            strawberry (strawberry)    lemon (lemon) } }
    bacteria (bacteria) {
        Escherichia_Coli (Escherichia Coli, E. Coli)
        Streptococcus_aureus (Streptococcus Aureus, S. Aureus)} }
```



Fig. 4. Ontology O_N . The degree of understanding that agent T has about O_N is 0.64 (Fig. 5)

On the other hand (figure 6), to find out the degree of understanding that agent N (with ontology O_N) has about ontology O_T of figure 2, we need to find $sim(c_T)$ for each $c_T \in O_T$, and to average their sv 's. Thus, $du(N, O_T) = (\Sigma sv)/|O_T| = 10.08/47 = 0.21$ is the degree of understanding that agent N has about ontology O_T . N knows less about O_T than T about O_N . The *understanding* of O_N by T increases as each sv increases and O_T grows. In more realistic ontologies, *relations* (such as *eats*, *part_of*, *lives_in*) are also nodes of the ontology, contributing to *du*.

¹⁷ B does not know how many concepts there are in O_A , so it needs cooperation from A, for instance, when B asks A “give me the next concept from your ontology.”

c_N	$c_T = \text{sim}(c_N)$	sv of sim
living-creature	living_creature	0.8
animal	animal	1
plant_creature	plant_creature	1
bacteria	bacteria	1
frog	frog	0.64
iguana	iguana	0.64
diprotodon	son_of animal	0.5
big_plant	son_of plant_creature	0.5
small_plant	son_of plant_creature	0.5
coconut	coconut	1
Escherichia Coli	-	0
Streptococcus Aureus	-	0
mango	mango	1
lemon	lemon	1
strawberry	-	0

Fig. 5. How well T knows O_N . Computing $\text{du}(T, O_N)$ is like asking T how much it knows about each concept c_N . The sv's in the last column are the answer. Adding these sv's and dividing into $|O_N|=15$, the degree of understanding of T with respect to O_N is found to be 0.64

3.3 Finding and Correcting the Source of a Disagreement

§3.1 shows that agent A can not perceive or see c_B directly. Given $c_A \in O_A$ and its most similar concept $c_B \in O_B$, can A perceive in what way c_B differs from its c_A ? After all, A knows from the value sv returned by $\text{sim}(c_A)$, how imperfect is the matching of c_B to c_A .

The answer is yes, and the following **Process P** computes it. Agent A can ask B about the relations in which c_B takes part [That is, arcs linking c_B with other concepts or with words or tokens¹⁰]. It will receive the answers in words. Then, A can process them (through sim) to see how c_A 's relations differ from those received. It can do the same with the father_of(c_B), and with the sons_of(c_B). And so on. Some words received will refer to relations of which A is not sure (it has no name for them, or there is ambiguity), so that more processing (Process P is called again) on these relations is needed. Sometimes, B will mention (words for) a concept in O_B of which A is not sure (so, Process P is called again) or is not in O_A . Occasionally agent A will receive from B assertions about c_B which A has as false for c_A .

An agent learning from another agent. For concept c_A , agent A can make a note N_{cA} containing what B knows about c_A which differs from what A knows about c_A : the values c_B , sv , and other differences between c_A and c_B . N_{cA} could be considered a belief about agent B: B believes N_{cA} about c_A , and N_{cA} is not what A knows about c_A . As one step further, agent A can *internalize* N_{cA} ; that is, own (believe, digest) N_{cA} : agent A can *learn* N_{cA} about c_A from O_B . For this to happen, agent A needs to incorporate the new relations and concepts (in N_{cA}) into its O_A , and to *resolve the ambiguities and inconsistencies* coming from N_{cA} (some of N_{cA} 's trios are known to A to be false [there is a contradiction]; others are ambiguous to A). This has been solved for an *agent teaching a person* but not yet for an agent teaching another agent. We have no solution now. It can be done, we think, by using other *knowledge services* in the Web to referee disagreements between O_A and O_B and help A decide *who is wrong* about what (the “what” is already captured in N_{cA}).

c_T	$C_N = \text{sim}(c_T)$	$sv \text{ of sim}$
living_creature	living_creature	0.8
animal	animal	1
invertebrate	son_of animal	0.5
vertebrate	son_of animal	0.5
Iguana	iguana	0.64
frog	frog	0.64
plant_creature	plant_creature	1
tropical_fruit	son_of plant_creature	0.5
coconut	coconut	1
mango	mango	1
citrus	son_of plant_creature	0.5
lemon	lemon	1
bacteria	bacteria	1

Fig. 6. How well N knows each concept c_T in ontology O_T ? Each answer (c_N , second column) yields a similarity value (last column); only sv 's $\neq 0$ are shown. The following concepts in O_T found no similar concept ($sv=0$) in O_N (N does not know them): thing, physical_object, insect, fly_animal, cockroach, flea, mollusk, reptile, lizard, batrachians, mammal, zebra, rodent, rat, domestic_rat, country_rat, mole_rodent, fox, cat, dog, lion, donkey, man, bird, chicken, duck, parrot, hawk, fish, orange, tangerine, rare_fruit, artificial_object, abstract_object

4 Conclusions

- Methods are given to allow interaction and understanding between agents with different ontologies, so that there is no need to agree first on a standard set of concept definitions. Given a concept and associated words, a procedure for finding the most

similar concept in another ontology is shown, with examples, as well as a measure of the degree of understanding between two agents. It remains to test our methods with large, vastly different, or practical ontologies.

- Work exposed is a step towards free will interactions among agents, perhaps strange to each other (footnote 4), needing to make sense of their utterances. It opposes current trends: (1) using canned interactions (footnote 5), (2) using agents written by the same group, or (3) following the same data exchange standards.
- Interaction through standards (trend 3 above) will dominate the market for some time. A standard ontology in a discipline is a good thing, although it feels rigid and archaic after a while.¹⁸ It is easier to follow standards than to be “flexible, uncompromising and willing to try to understand new concepts.” Irrespective of standardization, our approach allows agents to be flexible and have general ways of trying to understand what each has to say, specially new or unusual things.
- A *standard ontology for concept-sharing is not needed*; if one is built, it will always lag behind, since new concepts appearing every day will not be in it.

5 Suggestions for Further Work

Machine learning. Do the internalization of N_{cA} mentioned in §3.3; then, generalize to each $c_A \in O_A$ and somehow (we do not know how now) to each $c_B \in O_B$. This will allow agent A to learn (without human help) O_B from B. The new O_A = its old O_A merged with O_B . Alma-Delia Cuevas works along these lines.

Ontology merging. Is there a faster way for A to learn O_B in §3.3? Surely, agent A can get rid of its O_A and use (copy) O_B instead. This is too drastic: agent A forgets or erases what it already knows, in favor of B’s knowledge. Perhaps A can build O_B on top of A, and patch somehow O_A to accommodate for inconsistencies. *Suggestion:* use notes N_{cA} . This we have called *ontology merging*; more work is needed. Or there is the proposal by [16] to use *words* as concepts. *Improvement:* Add a crawler that combs the Web for ontologies suitable for merging.

Better notation for ontologies. • Tree notation is cumbersome, since only one subset relation is represented, and often a set S is partitioned into several partitions. Thus, a better notation could be:

```
person      partition sex (=M : male_person) (=F : female_person) }
              {partition age   (<=20 : young_person)
                (20< age <= 50 : adult_person)
                (>50 : old_person) }
```

- Similarly, graphs are cumbersome for representing n-ary relations. • When characterizing the relations (as another sub-tree of the ontology), you need to define types of partitioning relations (sex, age...), or whether the partition is a “natural” one, like partitioning vertebrate into fish, bird, reptile, batrachian and mammal.

¹⁸ Compare the UNESCO Catalog of Sciences (which is 30-years obsolete in Computer Science) with the ACM Computing Classification System, which is 2-years obsolete.

Agent interaction. Establish necessary or sufficient conditions for agent interaction lacking a communication agreement, as mentioned in §1.

Acknowledgments. Helpful discussions were held with Profs. Serguei Levachkine (CIC), Michael N. Huhns (U. of South Carolina), Alexander Gelbukh (CIC), and Hal Berghel (U. of Nevada Las Vegas). Work herein reported was partially supported by NSF-CONACYT Grant 32973-A and SNI-CONACYT *National Scientist* awards.

References

1. Victor Alexandrov, S Levachkine, Adolfo Guzman-Arenas. “*Data Dynamical Structures for Image Treatment with Applications to Digital Cartography*”. Book in preparation.
2. John Everett, D Bobrow, *et al* (2002) Making ontologies work for resolving redundancies across documents. *Comm. ACM* **45**, 2, 55-60. February.
3. K. Forbus, B. Falkenhainer, D. Gentner. (1989) The structure mapping engine: algorithms and examples. *Artificial Intelligence* **41**, 1, 1-63.
4. A. Gelbukh, G. Sidorov, and A. Guzman-Arenas. (1999) Document comparison with a weighted topic hierarchy. *DEXA-99, 10th International Conference on Database and Expert System applications*, Workshop on Document Analysis and Understanding for Document Databases, 566-570. Florence, Italy, August 30 to September 3.
5. T. Gruber. A translation approach to portable ontologies. *Knowledge* **7**.
6. Adolfo Guzman-Arenas (1998) Finding the main themes in a Spanish document. *Journal Expert Systems with Applications*, Vol. **14**, No. 1/2, 139-148, Jan./Feb. Elsevier.
7. A. Guzman-Arenas, C. Dominguez, and J. Olivares. (2002) Reacting to unexpected events and communicating in spite of mixed ontologies. *Lecture Notes in Artificial Intelligence* **2313**, 377-386, Carlos A. Coello *et al* (eds), Springer Verlag, Heidelberg.
8. Adolfo Guzman-Arenas and Jesus Olivares. (2004) Finding the Most Similar Concepts in two Different Ontologies. Accepted in *MICAI 04*. To appear as part of a *LNCS* book.
9. A. Guzman-Arenas and Serguei Levachkine. (2004) Hierarchies Measuring Qualitative Variables. *Lecture Notes in Computer Science* **2945**, 262-274, A. Gelbukh (ed).
10. Cloyd W. Holsapple and K. D. Joshi. (2002) A collaborative approach to ontology design. *Comm. ACM* **45**, 2, 42-47. February.
11. M. N. Huhns; M. P. Singh. and T. Ksiezyk (1997) Global Information Management Via Local Autonomous Agents. In: M. N. Huhns, Munindar P. Singh, (eds.): *Readings in Agents*, Morgan Kauffmann Publishers, Inc. San Francisco, CA
12. Henry Kim. (2002) Predicting how ontologies for the semantic web will evolve. *Comm. ACM* **45**, 2, 48-54. February.
13. Douglas B. Lenat, R. V. Guha, Karen Pittman, Dexter Pratt and Mary Shepherd (1990). Cyc: Toward Programs with Common Sense, *Comm. of the ACM* **33**, 9, 30 – 49.
14. M. Montes-y-Gomez, A. Lopez-Lopez, and A. Gelbukh. (2000) Information Retrieval with Conceptual Graph Matching. *Lecture Notes in Computer Science* **1873**, 312-321.
15. Jesus Olivares (2002) *An Interaction Model among Purposeful Agents, Mixed Ontologies and Unexpected Events*. Ph. D. Thesis, CIC-IPN. Mexico (In Spanish) Available on line at <http://www.jesusolivares.com/interaction/publica>
16. Yorick Wilks, B. Slator, L. Guthrie. (1996) *Electric words. Dictionaries, computers, and meanings*. ACL-MIT Press. Cambridge, USA. ISBN 0-262-23182-4 (hc)

Concept Indexing for Automated Text Categorization

José María Gómez¹, José Carlos Cortizo², Enrique Puertas¹, and Miguel Ruiz²

¹ Universidad Europea de Madrid

Villaviciosa de Odón, 28670 Madrid, Spain

{jmgomez, epuertas}@uem.es

² AINet Solutions

Fuenlabrada, 28943, Madrid, Spain

jccp@ainetsolutions.com, ethan113@hotmail.com

Abstract. In this paper we explore the potential of concept indexing with WordNet synsets for Text Categorization, in comparison with the traditional bag of words text representation model. We have performed a series of experiments in which we also test the possibility of using simple yet robust disambiguation methods for concept indexing, and the effectiveness of stoplist-filtering and stemming on the SemCor semantic concordance. Results are not conclusive yet promising.

1 Introduction

Automated Text Categorization (ATC) – the automated assignment of documents to predefined categories – is a text classification task that has attracted much interest in the recent years. Many types of documents (reports, books, Web pages, email messages, etc.) can be classified according to different kinds of categories, most often thematic or subject oriented: the Library of Congress Subject Headings, the categories directories like Yahoo!, etc. (see [1] for a survey).

While it is possible to build an ATC system by manually writing sets of classification rules, the most popular approach nowadays consist of using Information Retrieval (IR) and Machine Learning (ML) techniques to automatically induce a classification system called a *classifier* [1]. Such a classifier is obtained by: (1) representing (indexing) a set of manually classified documents (the *training collection*) as term weight vectors – as in the Salton’s Vector Space Model [2], and (2) learning a classification function that can be a set of rules, a decision tree, etc. This approach is quite effective for subject-based ATC, producing accurate classifiers provided enough training data is available.

An important decision in this learning-based model is the definition of what a term is. Most often, terms are defined as stoplist filtered, stemmed words. This may be sufficient for accurate learning, since individual words carry an important part of the meaning of text. However, this does not always hold, due to polysemy and synonymy of words. So, a number of term definitions have been tested in the bibliography, including word phrases obtained by statistical

or linguistic analysis (e.g. [3,4]), Information Extraction patterns (as in [5]), and WordNet sets of synonyms or *synsets* (e.g. [6,7]).

The latter indexing term definition is specially promising, according to some results obtained for IR tasks (see e.g. [8]). In this paper, we focus in using WordNet synsets as indexing units, in order to address the problems of synonymy and polysemy that are faced by text classification tasks. In the literature, a number of approaches have been tested, with mixed results [6,9,10,11,7]. To the date, however, there is no an in-depth study using a wide range of text representation options, feature selection methods, learning algorithms and problem definitions. This paper attempts to be such a study, in which we report experiments to test the hypothesis that concept indexing using WordNet synsets is a more effective text representation than using stoplist filtered, stemmed words as indexing terms. In our experiments, we have tested a number of text representations for a representative range of ML algorithms. Also, and since indexing with WordNet synsets involves a word sense disambiguation process, several disambiguation strategies have been tested.

We have organized this paper the following way. First, we introduce the most popular text representation approaches used in the literature. Then, we focus in conceptual indexing using WordNet synsets. The design of our experiments is after described, including the text representations and ML algorithms tested, the benchmark text collection, and the evaluation metrics. Then we present and analyze the results of our experiments, and finally our conclusions and future work are described.

2 Text Representation Approaches for Automated Text Categorization

Text classification tasks, such as IR, ATC, Text Filtering and Routing, and others, share the necessity of representing or indexing documents. It is not guaranteed that a representation method will be equally effective for all of them, but an improvement in text representation for TC may be of benefit for other tasks. In fact, some researchers support that TC is an excellent vehicle to studying different text representations. In words of Lewis ([4], page 72): “(...) We mentioned a number of problems with comparing text representations on a standard text retrieval test collection. (...) Text categorization as a vehicle for studying text representation addresses all (...) of this problems. (...) Of text classification tasks, text categorization seems best suited to studying text representation.”

In TC, documents are usually represented as term weight vectors, as in the Vector Space Model for IR [1]. In this model, called the *bag of words* representation in the literature, terms are most often defined as words stems, after filtering by using a stoplist, and applying a stemming algorithms like Porter’s. Weights can be binary (1 representing that a word stem occurs in the document, and 0 in the other case), TF (Term Frequency, the number of times that a word stem occurs in the document), or TF.IDF (in which IDF stands for Inverse Document Frequency, usually defined as $\log_2(n/df(t))$, being n the number of documents used

for learning, and $df(t)$ the number of documents in which the term t occurs). This term-weight (or in the vocabulary of ML, attribute-value) representation allows learning algorithms to be applied, obtaining a classification model called a classifier. Depending on the selected learning method, it can be needed to select a subset of the original terms¹ according to some quality metric, like Information Gain or χ^2 (see [1] for others).

The definition of term, or in other words, the indexing unit, is critical in TC. Terms must be good from a semantic point of view (that is, they must capture as much as possible the meaning of the texts), and from a learning point of view (that is, they must allow efficient and effective learning). A number of alternative definitions of term have been tested in the bibliography, including: (1) Character n-grams, that are short sequences of alphanumeric characters [12], appropriate for documents with OCR errors; (2) statistical and linguistic phrases (see e.g. [3, 4,7]), that is, multi-word expressions, either built using cooccurrence statistics in documents, or by shallow Natural Language Processing; its usage has not proven definitely superior to word stem based indexing, although it seems promising [3]; and (3) Information Extraction patterns, used in conjunction with the Relevancy Signatures algorithm by Riloff and others [5]; the patterns, called *signatures*, are $\langle word, semantic_node \rangle$ pairs in which words act as semantic node triggers, and these are defined for the domain at hand. This latter approach guarantees high precision TC, yet recall can be hurt. Some other approaches have been tested, but none has been clearly proven better than the bag of words representation, over a range of Machine Learning algorithms and a variety of application domains.

3 Concept Indexing with WordNet Synsets

The popularity of the bag of words model is justified by the fact that words and its stems carry an important part of the meaning of a text, specially regarding subject-based classification. However, this representation faces two main problems: the synonymy and the polysemy of words. These problems are addressed by a concept indexing model using WordNet synsets.

3.1 The Lexical Database WordNet

WordNet synsets are excellent candidates to index terms in text classification tasks, and allow addressing these problems. WordNet is a Lexical Database that accumulates lexical information about the words of the English language [13]. WordNet uses Synonym Sets or synsets as basic information and organization units. A synset contains a number of synsets that define a concept, which is one of the possible senses of the words or collocations in the synset. WordNet also stores information about lexical and conceptual relations between words, and concepts, including hyponymy or IS-A links, meronymy or HAS-A links, and others. This kind of information in WordNet makes it more a very populated semantic net and ontology than an electronic dictionary or thesaurus. Current contents of

¹ This process is often called “feature selection” in the literature.

WordNet (1.7.1) include more than 146,000 words and collocations and 111,000 synsets for nouns, verbs, adjectives and adverbs of the English language.

3.2 WordNet Synsets as Indexing Units for TC

The wide coverage of WordNet and its free availability has promoted its utilization for a variety of text classification tasks, including IR and TC². While WordNet usage for text classification has not proven widely effective (see e.g. [7, 14]), some works in which WordNet synsets are used as indexing terms for IR and TC are very promising [8,6,15,11]; see [16] for an in-depth discussion.

The basic idea of concept indexing with WordNet synsets is recognizing the synsets to which words in texts refer, and using them as terms for representation of documents in a Vector Space Model. Synset weights in documents can be computed using the same formulas for word stem terms in the bag of words representation. This concept based representation can improve IR, as commented by Gonzalo *et al.* [8]: “(...) using WordNet synsets as indexing space instead of word forms (...) combines two benefits for retrieval: one, that terms are fully disambiguated (this should improve precision); and two, that equivalent terms can be identified (this should improve recall).”

Experiments focused on concept indexing with WordNet synsets for TC have mixed results. On one side, lack of disambiguation has led to loss of effectiveness in some works. On the other, it is not clear that full disambiguation is absolutely required to obtain a document representation more effective than the bag of words model. We discuss three works specially relevant.

Scott and Matwin [7] have tested a text representation in which WordNet synsets corresponding to the words in documents, and their hypernyms, were used as indexing units with the rule learner Ripper on the Reuters-21578 test collection. The results of the experiments were discouraging, probably due to the fact that no disambiguation at all is performed, and to the inability of Ripper to accurately learn in a highly dimensional space.

Fukumoto and Suzuki [6] have performed experiments extracting synonyms and hypernyms from WordNet nouns in a more sophisticated fashion. First, synsets are not used as indexing units; instead, words extracted from synsets whose words occur in the documents are used. Second, the height to which the WordNet hierarchy is scanned is dependent on the semantic field (location, person, activity, etc.), and estimated during learning. These experiments were performed with Support Vector Machines on the Reuters-21578 test collection, and their results are positive, with special incidence on rare (low frequency) categories. Notably, no sense disambiguation was performed.

Petridis *et al.* [11] used WordNet synsets as indexing units with several learning algorithms on the Semcor text collection. In this collection, all words and collocations have been manually disambiguated with respect to WordNet synsets. The lazy learner k-Nearest Neighbors, the probabilistic approach Naive Bayes,

² See WordNet home page (<http://www.cogsci.princeton.edu/~wn/>) for a long bibliography.

and a Neural Network were tested on several text representations. The concept indexing approach performed consistently better than the bag of words model, being the Neural Network the best learner.

The work by Scott and Matwin suggests that some kind of disambiguation is required. The work by Fukumoto and Suzuki allows to suppose that no full disambiguation is needed. Finally, the work by Petridis *et al.* demonstrates that perfect disambiguation is effective, over a limited number of learning algorithms and an correctly disambiguated text collection.

However, to the date no experiments have been performed to test concept indexing over a representative range of learning algorithms and feature selection strategies. Also, the effect of of disambiguation remains to be tested. More in detail, current technologies are still far of perfect disambiguation (see the SEN-SEVAL conferences results [17]), and the discussion about the level of accuracy required for disambiguation to help IR [16]. However, there are simple but robust disambiguation strategies not yet tested in TC: disambiguation by part of speech – if a word has two or more possible part of speech categories (e.g. noun, verb, etc), only the senses for the right part of speech are taken³; and disambiguation by frequency – the most frequent sense (on a reference corpus⁴) for the correct part of speech is taken.

Our experiments are so focused on testing the potential of WordNet synsets as indexing units for TC, considering a representative range of feature selection strategies ans learning algorithms. Also, we compare several concept indexing strategies (with perfect disambiguation, disambiguation by part of speech, and by frequency) and bag of words representations.

4 The Design of Experiments

In this section, we describe our experiments setup, with special attention to the benchmark collection and evaluation metrics used in out work, and the text representation approaches and learning algorithms tested.

4.1 The SemCor Benchmark Collection

The SemCor text collection [18] is a Semantic Concordance, a corpus tagged with WordNet senses in order to supplement WordNet itself (for instance, for researching or showing examples of sense usage). However, SemCor has been adapted and used for testing IR by Gonzalo *et al.* [8], and used for evaluating TC by Petridis *et al.* [11]. Moreover, there are not other collections tagged with conceptual information in depth, and so, indexing with “perfect” disambiguation can hardly tested without SemCor.

SemCor is a subset of the Brown Corpus and the “The Red Badge of Courage” of about 250,000 words, in which each word or collocation is tagged

³ Note that several senses, and thus, synsets, are selected.

⁴ E.g. The corpus from which the senses were taken. WordNet senses are ordered according to their frequency in the reference corpus used building it.

with its correct WordNet sense using SGML. It is important to note that available information in SemCor allows both sense and concept indexing. As sense indexing, we understand using word senses as indexing units. For instance, we could use the pair (car, sense 1) or “car_s1” as indexing unit. Concept indexing involves a word-independent normalization that allows recognizing “car_s1” and “automobile_s1” as occurrences of the same concept, the noun of code 02573998 in WordNet (thus addressing synonymy and polysemy simultaneously).

SemCor needs not to be adapted for testing TC. It contains 352 text fragments obtained from several sources, and covering 15 text genres, like Press: Reportage, Religion, or Fiction: Science (see the full list in Table 1). We have used the first 186 text fragments (fully tagged) as testing documents, and the 15 genres as target categories. We must note that classes are not overlapping, being each document in exactly one category. Also, the classification problem is closer to genre than subject-based TC, although some genre differences are also related to the topics covered by the texts. Finally, the number of documents in each category is variable, from 2 (Press: Editorial) to 43 (Learned).

Given that classes are not overlapping, the TC problem is a multi-class problem (select the suitable category among $m = 15$ for a given document). However, some of the learning algorithms used in our work (e.g. Support Vector Machines) only allow binary class problems (classifying a document in a class or not). The SemCor multi-class TC problem can be transformed into $m = 15$ binary problems, in which a classifier is build for each class⁵. We have performed experiments on the binary problems for all algorithms, and in the multi-class only for those allowing it, as described below.

4.2 Evaluation Metrics

Effectiveness of TC approaches is usually measured using IR evaluation metrics, like recall, precision and F_1 [1]. We have used F_1 , more suitable for TC than others used in related work (like accuracy in [11]). This metric is an average of recall and precision, and can be averaged on the categories by macro- and micro-averaging. The first kind of average gives equal importance to all categories, while the second gives more importance to more populated categories. It is sensible to compute both. Macro- and micro-averaged F_1 values are noted as F_1^M and F_1^m respectively.

Often, TC test collections are divided into two parts: one for learning, and one for testing (the most prominent example is the Reuters-21578 test collection, with three consolidated partitions). When there is not a reference partition, or training data is limited, a k -fold cross validation process is used to estimate the evaluation metrics values. In short, the data is randomly divided in k groups (preserving class distribution), and k experiments are run using $k - 1$ groups for learning, and 1 for testing. The values obtained on each experiment are averaged on the k runs. Our test have been performed with 10-fold cross validation.

⁵ This kind of transformation is called “one against the rest” in the literature [11].

Table 1. Number of documents and indexing units per class (binary problem), for the seven text representation approaches tested, and selection with $IG > 0$. The last row shows the number of potential units.

Name	# docs	CD	CF	CA	BNN	BNS	BSN	BSS
Press: Reportage	7	845	536	1363	1133	840	1025	745
Press: Editorial	2	363	275	482	371	249	362	241
Press: Reviews	3	404	306	603	530	396	495	369
Religion	4	406	314	644	406	267	383	251
Skill and Hobbies	14	1487	1052	1879	1354	766	1258	683
Popular Lore	19	1540	1046	1455	1585	973	1494	895
Belles-Lettres	18	1482	1051	1604	1527	959	1428	876
Miscellaneous	12	1071	766	1658	1071	691	899	531
Learned	43	995	815	3091	1062	861	909	713
Fiction: General	29	714	596	2183	738	625	624	515
Fiction: Mystery	11	895	605	1288	822	567	738	484
Fiction: Science	2	296	240	415	326	230	280	194
Fiction: Adventure	10	1180	788	1791	1140	826	1031	722
Fiction: Romance	6	558	415	875	535	359	499	324
Humor	6	682	544	862	853	615	788	560
Initial (NOS)	186	25867	18780	37038	25728	16679	24188	15327

4.3 Text Representation Approaches

We have tested two kinds of text representation approaches for TC: a bag of words model, and a concept indexing model. Given that the TC problem is more oriented to genre than to subject, we have considered four possibilities: using a stoplist and stemming, using a stoplist, using stemming, and finally using words (without stoplist and stemming). These approaches are coded below as BSS, BSN, BNS, and BNN respectively. The motivation of considering these four approaches is that words occurring in a stoplist (e.g. prepositions, etc.) and original word forms (e.g. past suffixes as “-ed”) can be good indicators of different text genres (see e.g. [19]).

The three concept indexing approaches considered in our experiments are those regarding the level of disambiguation, which are: using the correct word sense (CD), using the first sense given the part of speech (CF), and using all the senses for the given part of speech (CA). We have weighted terms and synsets in documents using the popular TF.IDF formula from the Vector Space Model⁶.

Usually in TC, a feature selection process is applied in order to detect the most informative features (indexing units), and to decrease the problem dimensionality allowing efficient learning. An effective approach is selecting the top scored indexing units according to a quality metric, as Information Gain (IG). We have tested for selection approaches: no selection, selecting the top 1% terms or concepts, selecting the top 10% terms or concepts, and selecting those

⁶ The indexing process have been performed with the experimental package `ir.jar` by Mooney and colleagues (<http://www.cs.utexas.edu/users/mooney/ir-course/>).

Table 2. Number of indexing units per representation (multi-class problem), for each selection strategy.

Selection	CD	CF	CA	BNN	BNS	BSN	BSS
NOS	25867	18780	37038	25728	16679	24188	15327
S00	31	34	196	64	66	20	27

indexing units with IG score over 0. These approaches are coded below as NOS, S01, S10 and S00 respectively. The percentages are justified by other’s work [20].

In the Table 1, we show the resulting numbers of indexing units for each text representation approach and category, with S00 (which is class-dependent in the binary problems), and the original number of units (NOS) for each representation. Note there is not a strong correlation between the number of documents and units per class. For instance, the most populated classes (Learned and Fiction: General) have less indexing synsets than other less populated classes (e.g. Skill and Hobbies). Also, and as expected, the number of total indexing units is such as CA > CD > CF, and BNN > BSN > BNS > BSS.

Also, we report the number of documents per class. In the Table 2, we show the number of indexing units per representation in the multi-class problems. It is remarkable that the number of units when S00 ($IG > 0$) is applied, is quite smaller than in the binary problems; This suggests that it is harder to find e.g. a word able to inform (decide) on all classes as a whole, than in a binary classification problem.

4.4 Machine Learning Algorithms

It is important for our hypothesis to test a representative range of high performance learning algorithms. From those tested in the literature [1], we have selected the following ones⁷: the probabilistic approach Naive Bayes (NB); the rule learner PART [21]; the decision tree learner C4.5; the lazy learning approach k -Nearest Neighbors (k NN), used $k = 1, 2, 5$ neighbors in our experiments; the Support Vector Machines kernel method; and the AdaBoost meta-learner applied to Naive Bayes and C4.5. In our experiments, we have made use of the WEKA⁸ learning package, with default parameters for all algorithms except for k NN, where we selected weighted voting instead of majority voting, and AdaBoost, where we tried 10 and 20 iterations.

5 Results and Analysis

Our experiments involve computing a pair (F_1^M, F_1^m) per algorithm, feature selection approach, kind of indexing unit, and classification problem. Given that we consider three k values for k NN, 10 and 20 iterations for AdaBoost applied to

⁷ We exclude some references for brevity. Please find a sample at [1].

⁸ See <http://www.cs.waikato.ac.nz/ml/weka/>.

Table 3. Summary of best binary classification results, obtained with $k = 1$ for k NN, 10 iterations for ABNB and 20 for ABC4.5, S00 for NB and ABNB, and S01 for the rest of learning algorithms. Best F_1^M and F_1^m scores per algorithm are shown in boldface.

		NB	k NN	C4.5	PART	SVM	ABNB	ABC4.5
CD	macro	.631	.128	.253	.258	.502	.638	.262
	micro	.739	.224	.375	.403	.773	.759	.425
CF	macro	.554	.166	.184	.267	.461	.539	.228
	micro	.690	.304	.311	.431	.699	.680	.434
CA	macro	.611	.245	.219	.238	.567	.587	.226
	micro	.665	.409	.351	.427	.690	.680	.439
BNN	macro	.635	.130	.270	.308	.482	.638	.367
	micro	.750	.287	.382	.460	.730	.760	.526
BNS	macro	.536	.194	.245	.277	.538	.522	.290
	micro	.694	.347	.431	.479	.709	.682	.509
BSN	macro	.587	.116	.248	.262	.451	.577	.346
	micro	.722	.241	.351	.422	.685	.717	.531
BSS	macro	.494	.171	.198	.254	.518	.502	.296
	micro	.646	.325	.393	.459	.671	.649	.511

2 algorithms, and that SVM is only applied in the binary classification problems, these experiments produce $10 \times 4 \times 7 = 280$ binary classification F_1 pairs, and $9 \times 4 \times 7 = 252$ multi-class F_1 pairs. For brevity, we have summarized our experiments in Tables 3 and 4: the pair (F_1^M, F_1^m) for each learning algorithm/indexing approach is presented, according to the best selection metric.

Also, we have compared the number of times an indexing approach was better than others, focusing on answering: (a) Is conceptual indexing better than the bag of words model? (b) Are the simple disambiguation methods effective? and (c) Which is the impact of using stemming and a stoplist on effectiveness, in the bag of words model? We have compared the F_1 pairs for each algorithm and selection approach, leading to $11 \times 4 = 44$ comparisons for binary classification, and $10 \times 4 = 40$ comparisons for the multi-class problem. The results of these comparisons are summarized in the Table 5.

Examining these tables and the omitted results, and focusing on our research goals, we find that there is not a dominant indexing approach over all algorithms. Effectiveness of learning algorithms and feature selection methods is quite variable, partly motivated by the small size of training data. Remarkably, Naive Bayes, Support Vector Machines and AdaBoost with Naive Bayes perform comparably on the binary case, which is the most general and interesting one.

When comparing concept indexing and the bag of words model, we find that perfect disambiguated concept indexing (CD) outperforms all the bag of words approaches (B^*) on 20 of 84 times. Also, the highest F_1^M value are obtained with ABNB+CD or BNN (binary) and ABC45+BSN (multi-class), and the highest F_1^m are obtained with SVM+CD(binary) and ABNB+BNN (multi-class). In general, there is not a strong evidence of that concept indexing outperforms the bag of words model on the SemCor collection. The lack of training data,

Table 4. Summary of best multi-class classification results, obtained with $k = 1$ for k NN, 20 iterations for AB*, S00 for k NN, S10 for ABC4.5, and S01 for the rest of learning algorithms. Best F_1^M and F_1^m scores per algorithm are shown in boldface.

		NB	k NN	C4.5	PART	ABC4.5	ABNB
CD	macro	.319	.346	.261	.248	.339	.265
	micro	.446	.414	.403	.419	.468	.425
CF	macro	.274	.290	.244	.234	.282	.247
	micro	.368	.330	.373	.368	.427	.373
CA	macro	.232	.271	.199	.181	.297	.216
	micro	.317	.349	.290	.253	.441	.414
BNN	macro	.329	.281	.273	.254	.343	.321
	micro	.484	.366	.376	.403	.462	.489
BNS	macro	.291	.228	.224	.243	.277	.276
	micro	.425	.344	.323	.355	.419	.414
BSN	macro	.296	.292	.373	.244	.396	.277
	micro	.409	.349	.425	.333	.473	.446
BSS	macro	.253	.313	.281	.211	.297	.266
	micro	.355	.344	.301	.306	.414	.387

the skewness of class distribution, and the fact that other semantic relations in WordNet have been not used in the experiments, can explain these results.

Focusing on the disambiguation problem, we find that perfect disambiguation improves the two other approaches tested on 57 of 84 times, and rarely (except on k NN) has worst results than any of them on Tables 3 and 4. We believe that errors in disambiguation can have an important impact on performance, perhaps more than those reported by Gonzalo *et al.* [8]. Also, when testing the bag of words model on this genre-oriented classification task, we find that *not*

Table 5. Results of comparisons between (a) perfect disambiguation conceptual indexing vs. indexing with the two other disambiguation methods (CD > C*), (b) the bag of words model without stemming and stoplist vs. the other combinations (BNN > B*), and (c), the perfect disambiguation conceptual indexing vs. the bag of words model for all stoplist/stemming combinations (CD > B*).

	F_1^M	F_1^m	Both Of	
# CD > C* (Bin)	13	16	22	44
# CD > C* (MC)	31	33	35	40
# CD > C* (All)	44	49	57	84
# BNN > B* (Bin)	9	20	12	44
# BNN > B* (MC)	14	19	19	40
# BNN > B* (All)	23	39	31	84
# CD > B* (Bin)	6	12	9	44
# CD > B* (MC)	14	21	16	40
# CD > B* (All)	20	33	25	84

using stoplist and stemmer is better than other combinations 20 of 84 times, and the behavior on the figures reported on Tables 3 and 4 is rather erratic. We consider that a more in depth study of the application of genre oriented features (including e.g. tense, punctuation marks, etc.) is needed in this collection.

Regarding other's work, the experiments by Petridis *et al.* [11] are the most similar to ours, but far from comparable. This is due to the fact that they evaluate the learning algorithms on a different subset of Semcor, and by 3-fold cross validation, showing their results for a the less suitable evaluation measure: accuracy. However, their work has partially inspired ours.

6 Conclusions

In general, we have not been able to prove that concept indexing is better than the bag of words model for TC. However, our results must be examined with care, because of at least two reasons: (1) the lack of enough training data: stronger evidence is got when the number of documents increases; also, the behavior of some algorithms is surprising (Naive Bayes, k NN); and (2) the genre identification problem is such that meaning of used words is not more relevant than other text features, including structure, capitalization, tense, punctuation, etc. In other words, this study needs to be extended to a more populated, subject-oriented TC test collection (e.g. Reuters-21578). The work by Petridis *et al.* [11] adds evidence of concept indexing outperforming the bag of words model on the SemCor collection, specially with SVM, and Fukumoto and Suzuki [6] work with them on Reuters-21578 allow to say that such an study is well motivated and promising.

We also plan to use semantic relations in WordNet in our future experiments on the Semcor and Reuters test collections. The results published in literature (e.g. [6]) suggest that using a limited amount of this information can improve our results in concept indexing.

ATC is a different task to IR. We think that, being TC categories more static than IR queries, and given that it is possible to apply ML feature selection strategies in TC, it is more likely that using concept indexing in TC will outperform the bag of words model, against Voorhees findings in IR [14].

References

1. Sebastiani, F.: Machine learning in automated text categorization. *ACM Computing Surveys* **34** (2002) 1–47
2. Salton, G.: Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison Wesley (1989)
3. Caropreso, M., Matwin, S., Sebastiani, F.: A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In: *Text Databases and Document Management: Theory and Practice*. Idea Group Publishing (2001) 78–102
4. Lewis, D.D.: Representation and learning in information retrieval. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, US (1992)

5. Riloff, E.: Using learned extraction patterns for text classification. In: Connectionist, statistical, and symbolic approaches to learning for natural language processing, Springer Verlag (1996) 275–289
6. Fukumoto, F., Suzuki, Y.: Learning lexical representation for text categorization. In: Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources. (2001)
7. Scott, S.: Feature engineering for a symbolic approach to text classification. Master's thesis, Computer Science Dept., University of Ottawa, Ottawa, CA (1998)
8. Gonzalo, J., Verdejo, F., Chugur, I., Cigarrán, J.: Indexing with WordNet synsets can improve text retrieval. In: Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems. (1998)
9. Junker, M., Abecker, A.: Exploiting thesaurus knowledge in rule induction for text classification. In: Proceedings of the, 2nd International Conference on Recent Advances in Natural Language Processing. (1997) 202–207
10. Liu, J., Chua, T.: Building semantic perceptron net for topic spotting. In: Proceedings of 37th Meeting of Association of Computational Linguistics. (2001)
11. Petridis, V., Kaburlasos, V., Fragkou, P., Kehagias, A.: Text classification using the σ -FLNMAP neural network. In: Proceedings of the 2001 International Joint Conference on Neural Networks. (2001)
12. Cavnar, W., Trenkle, J.: N-gram-based text categorization. In: Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US (1994) 161–175
13. Miller, G.A.: WordNet: A lexical database for English. Communications of the ACM **38** (1995) 39–41
14. Voorhees, E.: Using WordNet for text retrieval. In: WordNet: An Electronic Lexical Database. MIT Press (1998)
15. Mihalcea, R., Moldovan, D.: Semantic indexing using WordNet senses. In: Proceedings of ACL Workshop on IR and NLP. (2000)
16. Stokoe, C., Oakes, M.P., Tait, J.: Word sense disambiguation in information retrieval revisited. In: Proceedings of the 26th ACM International Conference on Research and Development in Information Retrieval. (2003)
17. Kilgarriff, A., Rosenzweig, J.: Framework and results for english SENSEVAL. Computers and the Humanities **34** (2000) 15–48
18. Miller, G.A., Leacock, C., Tengi, R., Bunker, R.: A semantic concordance. In: Proc. Of the ARPA Human Language Technology Workshop. (1993) 303–308
19. Kessler, B., Numberg, G., Schütze, H.: Automatic detection of text genre. In: Proceedings of ACL-97, 35th Annual Meeting of the Association for Computational Linguistics, Madrid, ES (1997) 32–38
20. Yang, Y., Pedersen, J.: A comparative study on feature selection in text categorization. In: Proc. Of the 14th International Conf. On Machine Learning. (1997)
21. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In Shavlik, J., ed.: Machine Learning: Proceedings of the Fifteenth International Conference, San Francisco, CA, Morgan Kaufmann Publishers (1998)

Acquiring Selectional Preferences from Untagged Text for Prepositional Phrase Attachment Disambiguation*

Hiram Calvo¹ and Alexander Gelbukh^{1,2}

¹ Center for Computing Research, National Polytechnic Institute,
Av. Juan de Dios Bátiz s/n, esq. Av. Mendizábal, México, D.F., 07738. México
hcalvo@sagitario.cic.ipn.mx,
gelbukh @ gelbukh.com; www.gelbukh.com

² Department of Computer Science and Engineering, Chung-Ang University,
221 Huksuk-Dong, DongJak-Ku, Seoul, 156-756, Korea

Abstract. Extracting information automatically from texts for database representation requires previously well-grouped phrases so that entities can be separated adequately. This problem is known as prepositional phrase (PP) attachment disambiguation. Current PP attachment disambiguation systems require an annotated treebank or they use an Internet connection to achieve a precision of more than 90%. Unfortunately, these resources are not always available. In addition, using the same techniques that use the Web as corpus may not achieve the same results when using local corpora. In this paper, we present an unsupervised method for generalizing local corpora information by means of semantic classification of nouns based on the top 25 unique beginner concepts of WordNet. Then we propose a method for using this information for PP attachment disambiguation.

1 Introduction

Extracting information automatically from texts for database representation requires previously well-grouped phrases so that entities can be separated adequately. For example in the sentence *See the cat with a telescope*, two different groupings are possible: *See [the cat] [with a telescope]* or *See [the cat with a telescope]*. The first case involves two different entities, while the second case has a single entity. This problem is known in syntactic analysis as prepositional phrase (PP) attachment disambiguation.

There are several methods to disambiguate a PP attachment. Earlier methods, e.g. those described in [1, 2], showed that up to 84.5% of accuracy could be achieved using treebank statistics. Kudo and Matsumoto [3] obtained 95.77% accuracy with an algorithm that needed weeks for training, and Lüdtke and Sato [4] achieved 94.9% accuracy requiring only 3 hours for training. These methods require a corpus annotated syntactically with chunk-marks. This kind of corpora is

* Work done under partial support of Mexican Government (CONACyT, SNI, PIFI-IPN, CGEPI-IPN), Korean Government (KIPA), ITRI of Chung-Ang University, and RITOS-2. The second author is currently on Sabbatical leave at Chung-Ang University.

Table 1. Occurrence examples for some verbs in Spanish

Triplet	Literal translation	English occurrences	% of total verb occurrences
ir a {actividad}	go to {activity}	711	2.41%
ir a {tiempo}	go to {time}	112	0.38%
ir hasta {comida}	go until {food}	1	0.00%
beber {sustancia}	drink {substance}	242	8.12%
beber de {sustancia}	drink of {substance}	106	3.56%
beber con {comida}	drink with {food}	1	0.03%
amar a {agente_causal}	love to {causal_agent}	70	2.77%
amar a {lugar}	love to {place}	12	0.47%
amar a {sustancia}	love to {substance}	2	0.08%

not available for every language, and the cost to build them can be relatively high, considering the number of person-hours that are needed. A method that works with untagged text is presented in [5]. This method has an accuracy of 82.3, it uses the Web as corpus and therefore it can be slow—up to 18 queries are used to resolve a single PP attachment ambiguity, and each preposition + noun pair found in a sentence multiplies this number.

The algorithm presented in [5] is based on the idea that a very big corpus has enough representative terms that allow PP attachment disambiguation. As nowadays it is possible to have locally very big corpora, we ran experiments to explore the possibility of applying such method without the limitation of an Internet connection. We tested with a very big corpus of 161 million words in 61 million sentences. This corpus was obtained on-line from 3 years of publication of 4 newspapers. The results were disappointing—the same algorithm that used the Web as corpus yielding a recall of almost 90% had a recall of only 36% with a precision of almost 67% using the local newspaper corpus.

Therefore, our hypothesis is that we need to generalize the information contained in the local newspaper corpus to maximize recall and precision. A way for doing this is using selectional preferences: a measure of the probability of a complement to be used for certain verb, based on the semantic classification of the complement. This way, the problem of analyzing *I see the cat with a telescope* can be solved by considering *I see {animal} with {instrument}* instead.

{food}:	breakfast, feast, cereal, beans, milk, etc.
{activity}:	abuse, education, lecture, fishing, hurry, test
{time}:	dawn, history, Thursday, middle age, childhood
{substance}:	alcohol, coal, chocolate, milk, morphine
{name}:	John, Peter, America, China
{causal_agent}:	lawyer, captain, director, intermediary, grandson
{place}:	airport, forest, pit, valley, courtyard, ranch

Fig. 1. Examples of words for categories shown in Table 1

Table 2. Examples of Semantic Classifications of Nouns

Word	English translation	Classification
rapaz	predatory	activity
rapidez	quickness	activity
rapiña	prey	shape
rancho	ranch	place
raqueta	racket	thing
raquitismo	rickets	activity
rascacielos	skyscraper	activity
rasgo	feature	shape
rastreo	tracking	activity
rastro	track	activity
rata	rat	animal
ratero	robber	causal agent
rato	moment	place
ratón	mouse	animal
raya	{ boundary manta ray dash	activity animal shape
rayo	ray	activity
raza	race	grouping
razón	reason	attribute
raíz	root	part
reacción	reaction	activity
reactor	reactor	thing
real	real	grouping
realidad	reality	attribute
realismo	realism	shape
realización	realization	activity
realizador	producer	causal agent

For example, to disambiguate the PP attachment for the Spanish sentence *Bebe de la jarra de la cocina* '(he) drinks from the jar of the kitchen' selectional preferences provide information such as *from {place}* is an uncommon complement for the verb *bebe* 'drinks', and thus, the probability of attaching this complement to the verb *bebe*, is low. Therefore, it is attached to the noun *jarra* yielding *Bebe de [la jarra de la cocina]* '(he) drinks [from the jar of the kitchen]'

Table 1 shows additional occurrence examples for some verbs in Spanish. From this table it can be seen that the verb *ir* 'to go' is mainly used with the complement *a {activity}* 'to {activity}'. Less used combinations have almost zero occurrences, such as *ir hasta {food}* lit. 'go until food'. The verb *amar* 'to love' is often used with the preposition *a* 'to'.

In this paper, we propose a method to obtain selectional preferences information such as that shown in Table 1. In Section 2, we will discuss briefly related work on selectional preferences. Sections 3 to 5 explain our method. In Section 6,

we present an experiment and evaluation of our method applied to PP attachment disambiguation, and finally we conclude the paper.

2 Related Work

The terms *selectional constraints* and *selectional preferences* are relatively new, although similar concepts are present in works such as [6] or [7]. One of the earliest works using these terms was [8], where Resnik considered selectional constraints to determine the restrictions that a verb imposes on its object. Selectional constraints have rough values, such as whether an object of certain type can be used with a verb. Selectional preferences are graded and measure, for example, the probability that an object can be used for some verb [9]. Such works use a shallow parsed corpus and a semantic class lexicon to find selectional preferences for word sense disambiguation.

Another work using semantic classes for syntactic disambiguation is [10]. In this work, Prescher *et al.* use an EM-Clustering algorithm to obtain a probabilistic lexicon based in classes. This lexicon is used to disambiguate target words in automatic translation.

A work that particularly uses WordNet classes to resolve PP attachment is [2]. In this work, Brill and Resnik apply the Transformation-Based Error-Driven Learning Model to disambiguate the PP attachment, obtaining an accuracy of 81.8%. This is a supervised algorithm.

As far as we know, selectional preferences have not been used in unsupervised models for PP attachment disambiguation.

3 Sources of Noun Semantic Classification

A semantic classification for nouns can be obtained from existing WordNets, using a reduced set of classes corresponding to the unique beginners for WordNet nouns described in [11]. These classes are: activity, animal, life_form, phenomenon, thing, causal_agent, place, flora, cognition, process, event, feeling, form, food, state, grouping, substance, attribute, time, part, possession, and motivation. To these unique beginners, *name* and *quantity* are added. Name corresponds to capitalized words not found in the semantic dictionary and Quantity corresponds to numbers.

Since not every word is covered by WordNet and since there is not a WordNet for every language, the semantic classes can be alternatively obtained automatically from Human-Oriented Explanatory Dictionaries. A method for doing this is explained in detail in [12]. Examples of semantic classification of nouns extracted from the human-oriented explanatory dictionary [13] using this method are shown in Table 2.

4 Preparing Sources for Extracting Selectional Preferences

Journals or newspapers are common sources of great amounts of medium to good quality text. However, usually these media exhibit a trend to express several ideas in little space.

This causes sentences to be long and full of subordinate sentences, especially for languages in which an unlimited number of sentences can be nested. Because of this, one of the first problems to be solved is to break a sentence into several sub-sentences. Consider for example the sentence shown in Figure 2—it is a single sentence, extracted from a Spanish newspaper.

Y ahora, cuando
 (el mundo) **está** gobernado por (las leyes del mercado), cuando
 (lo determinante en la vida) **es**
comprar o
vender, sin
fijarse en <los que
carecen de todo>,
 son fácilmente **comprensibles** <las razones de
 <la ola de publicidad global que
convenció <a los posibles compradores de servicios y regalos > de que
había (grandes razones) para
celebrar> y
 como les **pareciese** poco (el fin de año)
 se **lanzaron a**
propagar (el fin del siglo y del milenio)

Literal English translation:

And now, when
 the world is governed by market's laws, when
 what **determines life is**
to buy or
to sell without
taking into account those that
 don't **have** anything,
 easily understandable **are** the reasons for
 the global publicity wave that
convinced the possible buyers of services and gifts that
there were great reasons to
celebrate, and
 as the end of the year **was** not enough for them,
 they **launched** themselves
to propagate the end of the century and the millennium

Fig. 2. Example of a very long sentence in a style typically found in journals. () surround simple NPs; <> surround NP subordinate clauses, **verbs** are in boldface.

PREP V ,	CONJ PRON V	CONJ N V
V ADV que	PREP DET que N	PREP DET V
, PRON V	N que V	, N V
V PREP N , N V	, donde	N , que V
V PREP N , N PRON V	N , N	N , CONJ que
V PREP N V	CONJ N N V	N que N PRON V
V de que	CONJ N PRON V	CONJ PRON que V V

Fig. 3. Delimiter patterns. V: verb, PREP: preposition, CONJ: conjunction, DET: determiner, N: noun; lowercase are strings of words

We use two kinds of delimiters to separate subordinate sentences: delimiter words and delimiter patterns. Examples of delimiter words are *pues* ‘well’, *ya que* ‘given that’, *porque* ‘because’, *cuando* ‘when’, *como* ‘as’, *si* ‘if’, *por eso* ‘because of that’, *y luego* ‘and then’, *con lo cual* ‘with which’, *mientras* ‘in the meantime’, *con la cual* ‘with which’ (feminine), *mientras que* ‘while’. Examples of delimiter patterns are shown in Figure 3. These patterns are POS based, so the text was shallow-parsed before applying them.

The sentence in Figure 2 was separated using this simple technique so that each sub-sentence lies in a different row.

5 Extracting Selectional Preferences Information

Now that sentences are tagged and separated, our purpose is to find the following syntactic patterns:

1. Verb _{NEAR} Preposition _{NEXT_TO} Noun
2. Verb _{NEAR} Noun
3. Noun _{NEAR} Verb
4. Noun _{NEXT_TO} Preposition _{NEXT_TO} Noun

Patterns 1 to 3 will be referred henceforth as *verb patterns*. Pattern 4 will be referred as a *noun or noun classification pattern*. The _{NEAR} operator implies that there might be other words in-between. The operator _{NEXT_TO} implies that there are no words in-between. Note that word order is preserved, thus pattern 2 is different of pattern 3. The results of these patterns are stored in a database. For verbs, the lemma is stored. For nouns, its semantic classification, when available through Spanish WordNet, is stored. As a noun may have several semantic classifications, due to, for example, several word senses, a different pattern is stored for each semantic classification. For example, see Table 3. This table shows the information extracted for the sentence of Figure 2.

Once this information is collected, the occurrence of patterns is counted. For example, the last two rows in Table 3, *fin, de, año* and *fin, de, siglo* add 2 of each of the following occurrences: place of cognition, cognition of cognition, event of

Table 3. Semantic patterns information extracted from Sentence in Figure 2

Words	Literal translation	Pattern
<i>gobernado, por, ley</i>	governed, by, law	<i>gobernar, por,</i> cognition
<i>gobernado, de, mercado</i>	governed, of, market	<i>gobernar, de,</i> activity thing
<i>es, en, vida</i>	is, in, life	<i>ser, en,</i> state life_form
<i>causal_agent</i>		causal_agent attribute
<i>convenció, a, comprador</i>	convinced, to, buyer	<i>convencer, a,</i> causal_agent
<i>convenció, de, servicio</i>	convinced, of, service	<i>convencer, de,</i> activity process
		possession thing
		grouping
<i>pareciese, de, año</i>	may seem, of, year	<i>parecer, de,</i> cognition time
<i>lanzaron, de, año</i>	released, of, year	<i>lanzar, de,</i> cognition time
<i>propagar, de, siglo</i>	propagate, of, century	<i>propagar, de,</i> cognition time
<i>propagar, de, milenio</i>	propagate, of, millennium	<i>propagar, de,</i> cognition time
<i>ley, de, mercado</i>	law, of, market	cognition, <i>de,</i> activity thing
<i>ola, de, publicidad</i>	wave, of, publicity	event, <i>de,</i> activity cognition
<i>comprador, de, servicio</i>	buyer, of, service	causal_agent, <i>de,</i> activity process
		possession thing
		grouping
<i>fin, de, año</i>	end, of, year	place cognition event time, <i>de,</i> cognition time
<i>fin, de, siglo</i>	end, of, century	place cognition event time, <i>de,</i> cognition time

cognition, time of cognition, place of time, cognition of time, event of time, and time of time. An example of the kind of information that results from this process is shown in Table 1. This information is used then as a measure of the selectional preference that a noun has to a verb or to another noun.

6 Experimental Results

The procedure explained in the previous sections was applied to a corpus of 161 million words comprising more than 3 years of articles from four different Mexican newspapers. It took approximately three days on a Pentium IV PC to obtain 893,278 different selectional preferences for verb patterns (patterns 1 to 3) for 5,387 verb roots, and 55,469 different semantic selectional preferences for noun classification patterns (pattern 4).

6.1 PP Attachment Disambiguation

In order to evaluate the quality of the selectional preferences obtained, we tested them on the task of PP attachment disambiguation. Consider the first two rows of Table 4, corresponding to the fragment of text *governed by the laws of the market*. This fragment reported two selectional preferences patterns: *govern by {cognition}* and *govern of {activity/thing}*. With the selectional preferences obtained, it is possible to determine automatically the correct PP attachment: values of co-occurrence for *govern of {activity/thing}* and *{cognition} of {activity/thing}* are compared. The highest one sets the attachment.

Formally, to decide if the noun N_2 is attached to its preceding noun N_1 or is attached to the verb V of the local sentence, the values of frequency for these attachments are compared using the following formula [14]:

$$freq(X, P, C_2) = \frac{occ(X, P, C_2)}{occ(X) + occ(C_2)}$$

where X can be V , a verb, or C_1 , the classification of the first noun N_1 . P is a preposition, and C_2 is the classification of the second noun N_2 . If $freq(C_1, P, C_2) > freq(V, P, C_2)$, then the attachment is decided to the noun N_1 . Otherwise, the attachment is decided to the verb V . The values of $occ(X, P, C_2)$ are the number of occurrences of the corresponding pattern in the corpus. See Table 1 for examples of verb occurrences. Examples of noun classification occurrences taken from the Spanish journal corpus are: *{place} of {cognition}*: 354,213, *{place} with {food}*: 206, *{place} without {flora}*: 21. The values of $occ(X)$ are the number of occurrences of the verb or the noun classification in the corpus. For example, for *{place}* the number of occurrences is 2,858,150.

Table 4. Results of the PP attachment disambiguation using selectional preferences

file	#sentences	words	average words per sentence	kind of text	precision	recall
n1	252	4,384	17.40	news	80.76%	75.94%
t1	74	1,885	25.47	narrative	73.01%	71.12%
d1	220	4,657	21.17	sports	80.80%	81.08%
total: 546		10,926		average: 78.19%		76.04%

6.2 Evaluation

The evaluation was carried on 3 different files of LEXESP corpus [15], containing 10,926 words in 546 sentences. On average, this method achieved a precision of 78.19% and a recall of 76.04%. Details for each file processed are shown in Table 4.

7 Conclusions and Future Work

Using selectional preferences for PP attachment disambiguation yielded a precision of 78.19% and a recall of 76.04%. These results are not as good as the ones obtained with other methods, such as an accuracy of 95%. However, this method does not require any costly resource such as an annotated corpus, nor an Internet connection (using the web as corpus); it does not even need the use of a semantic hierarchy (such as WordNet), as the semantic classes can be obtained from Human-Oriented Explanatory Dictionaries, as it was discussed in Section 3.

We found also that, at least for this task, applying techniques that use the Web as corpus to local corpora reduces the performance of these techniques in more than 50%, even if the local corpora are very big.

In order to improve results for PP attachment disambiguation using selectional preferences, our hypothesis is that instead of using only 25 fixed semantic classes, intermediate classes can be obtained by using a whole hierarchy. In this way, it would be possible to have a flexible particularization for terms commonly used together, i.e. collocations, such as *fin de año* ‘end of year’, while maintaining the power of generalization. Another point of further developments is to add a WSD module, so that not every semantic classification for a single word is considered, as it was described in Section 5.

References

1. Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. A Maximum Entropy Model for Prepositional Phrase Attachment. In *Proceedings of the ARPA Human Language Technology Workshop*, 1994, pp. 250-255
2. Eric Brill and Philip Resnik. A Rule-Based Approach To Prepositional Phrase Attachment Disambiguation, In *Proceedings of COLING-1994*, 1994.
3. T. Kudo and Y. Matsumoto. Use of Support Vector Learning for Chunk Identification. In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 2000
4. Dirk Lüdtke and Satoshi Sato. Fast Base NP Chunking with Decision Trees - Experiments on Different POS Tag Settings. In Gelbukh, A. (ed) *Computational Linguistics and Intelligent Text Processing*, Springer LNCS, 2003, pp. 136-147
5. Hiram Calvo and Alexander Gelbukh. Improving Prepositional Phrase Attachment Disambiguation Using the Web as Corpus, In A. Sanfeliu and J. Shulcloper (Eds.) *Progress in Pattern Recognition*, Springer LNCS, 2003, pp. 604-610
6. Weinreich, Uriel. *Explorations in Semantic Theory*, Mouton, The Hague, 1972.
7. Dik, Simon C., *The Theory of Functional Grammar. Part I: The structure of the clause*. Dordrecht, Foris, 1989.
8. Philip Resnik. Selectional Constraints: An Information-Theoretic Model and its Computational Realization, *Cognition*, 61:127-159, November, 1996.

9. Philip Resnik. Selectional preference and sense disambiguation, presented at the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?, held April 4-5, 1997 in Washington, D.C., USA in conjunction with ANLP-97.
10. Detlef Prescher, Stefan Riezler, and Mats Rooth. Using a probabilistic class-based lexicon for lexical ambiguity resolution. In *Proceedings of the 18th International Conference on Computational Linguistics*, Saarland University, Saarbrücken, Germany, July-August 2000. ICCL.
11. Miller, G. WordNet: An on-line lexical database, In *International Journal of Lexicography*, 3(4), December 1990, pp. 235-312
12. Calvo, H. and A. Gelbukh. Extracting Semantic Categories of Nouns for Syntactic Disambiguation from Human-Oriented Explanatory Dictionaries, In Gelbukh, A. (ed) *Computational Linguistics and Intelligent Text Processing*, Springer LNCS, 2004.
13. Lara, Luis Fernando. *Diccionario del español usual en México*. Digital edition. Colegio de México, Center of Linguistic and Literary Studies, 1996.
14. Volk, Martin. Exploiting the WWW as a corpus to resolve PP attachment ambiguities. In *Proceeding of Corpus Linguistics 2001*. Lancaster: 2001
15. Sebastián, N., M. A. Martí, M. F. Carreiras, and F. Cuetos. *Lexesp, léxico informatizado del español*, Edicions de la Universitat de Barcelona, 2000

Semantic Enrichment for Ontology Mapping

Xiaomeng Su* and Jon Atle Gulla

Department of Computer and Information Science,
Norwegian University of Science and Technology,
N-7491, Trondheim, Norway
[{xiaomeng, jag}@idi.ntnu.no](mailto:{xiaomeng,jag}@idi.ntnu.no)

Abstract. In this paper, we present a heuristic mapping method and a prototype mapping system that support the process of semi-automatic ontology mapping for the purpose of improving semantic interoperability in heterogeneous systems. The approach is based on the idea of semantic enrichment, i.e. using instance information of the ontology to enrich the original ontology and calculate similarities between concepts in two ontologies. The functional settings for the mapping system are discussed and the evaluation of the prototype implementation of the approach is reported.

1 Introduction

System interoperability is an important issue, widely recognized in information technology intensive enterprises and in the research community of information systems (IS). An increasing dependence on and cooperation between organizations have created a need for many organizations to access remote as well as local information sources. The widely adoption of the World Wide Web to access and distribute informations further stresses the need for systems interoperability.

The current World Wide Web has well over 4.3 billion pages [9], but the vast majority of them are in human readable format only. In order to allow software agents to understand and process the web information in a more intelligent way, researchers have created the Semantic Web vision [3], where data has structure and ontologies describe the semantics of the data. The Semantic Web offers a compelling vision, yet it also raises many difficult challenges. One of the key challenges is to find semantic correspondences between ontologies.

Ontology is a key factor for enabling interoperability in the semantic web [3]. Ontologies are central to the semantic web because they allow applications to agree on the terms that they use when communicating. It facilitates communication by providing precise notions that can be used to compose messages (queries, statements) about the domain. For the receiving party, the ontology helps to understand messages by providing the correct interpretation context. Thus, ontologies, if shared among stakeholders, may improve system interoperability across ISs in different organizations and domains.

* Part of the research has been supported by Accenture, Norway

However, it is hard to come up with one single universal shared ontology, which is applauded by all players. It seems clear that ontologies face the same or even harder problems with respect to heterogeneity as any other piece of information [23]. The attempts to improve system interoperability will therefore rely on the reconciliation of different ontologies used in different systems.

So, interoperability among applications in heterogeneous systems depends critically on the ability to map between their corresponding ontologies. Today, matching between ontologies is still largely done by hand, in a labor-intensive and error-prone process [16]. As a consequence, semantic integration issues have now become a key bottleneck in the deployment of a wide variety of information management applications.

2 Overview of the Approach

In this section, we give a brief introduction of our approach. For a more comprehensive account of the approach, we refer to [22].

2.1 Prerequisite

The word ontology has been used to describe artifacts with different degrees of structure. These range from simple taxonomies (such as the Yahoo! hierarchy), to metadata schemes (such as the Dublin Core [6]), to logical theories. In our context: an *ontology*¹ specifies a conceptualization of a domain in terms of concepts, attributes and relations. Concepts are typically organized into a tree structure based on subsumption relationship among concepts. Ad hoc relations further connect concepts and formulate a semantic net structure in the end. We focus on finding mappings between concepts and between relations. This is because they are central components of ontologies and matching them successfully would aid in matching the rest of the ontologies. The ontologies express overlapping knowledge in a common domain and are represented in the Referent Modeling Language (RML) [20]. This is an Extended ER-like (Entity Relationship) graphical language with strong abstraction mechanism and a sound formal basis.

The overall process of ontology mapping can then be defined as: given two ontologies O_a and O_b , mapping one ontology to another means that for each element in ontology O_a , we find corresponding element(s), which has same or similar semantics, in ontology O_b and vice versa.

2.2 Idea Illustration

Ontology mapping concerns the interpretation of models of a Universe of Discourse (UoD), which in their turn are interpretations of the UoD. There is no

¹ We use the term ontology and concept model interchangeably in the rest of the paper unless otherwise explicitly specified

argumentation for these interpretations to be the only existing or complete conceptualizations of the state of affairs in the real world. We assume that the richer a description of a UoD is, the more accurate conceptualization we achieve of the same UoD through interpretation of the descriptions.

Hence, the starting point for comparing and mapping heterogeneous semantics in ontology mapping is to semantically enrich the ontologies. Semantic enrichment facilitates ontology mapping by making explicit different kinds of "hidden" information concerning the semantics of the modeled objects. The underlying assumption is that the more semantics that is explicitly specified about the ontologies, the more feasible their comparison becomes. We base our approach on extension analysis, i.e. instance information a concept possesses. The instances are documents that have been classified to the concepts. The idea behind is that written documents that are used in a domain inherently carry the conceptualizations that are shared by the members of the community. This approach is in particular attractive on the World Wide Web, because huge amounts of free text resources are available.

On the other hand, we consider information retrieval (IR) techniques as a vital component of our approach. With information retrieval, a concept node in the first ontology is considered as a query to be matched against the collection of concept nodes in the second ontology. Ontology mapping thus becomes a question of finding concept nodes from the second ontology that best relate to the query node. It is natural to think of representing the instance information in vectors, where the documents under one concept become building material for the vector of that concept. Therefore, the concept is semantically enriched with a generalization of the information its instances provide. The generalization takes the form of a high-dimensional vector. These concept vectors are ultimately used to compute the *degree of similarity* between pairs of concepts. The vector model allows us to discover concept pairs which matches only partially and the numerical degree of similarity gives a way to rank the matches.

2.3 System Functional View

In this section, we discuss the major steps of our method in more detail. A prototype system iMapper has been implemented to verify the applicability of the approach. Figure 1 illustrates the steps and their executing sequence. We will explain each step in turn. The two ontologies, together with the document sets are loaded into the system and let the algorithm run upon them to derive mappings between the two ontologies.

Document assignment. We use a linguistic based classifier CnS (Classification and Search) [4] to associate documents with the ontologies. It is a semi-automatic process, where users can adjust the automatic classification results. The assigning of documents to concept nodes is necessary when no instance knowledge of the ontology is available. However, if documents have already been

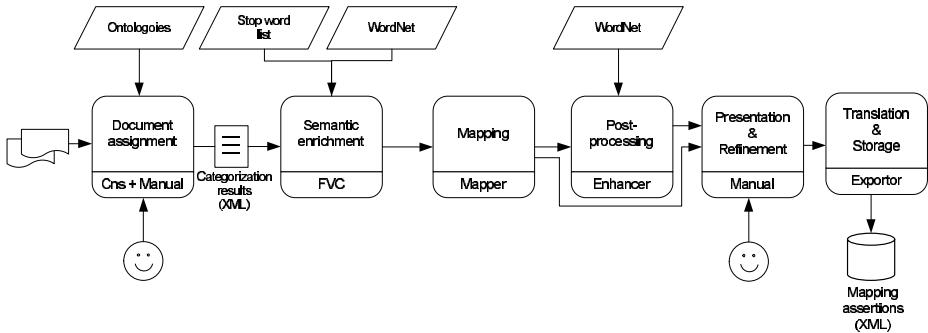


Fig. 1. Major steps of the approach

assigned to specific nodes, we can skip the first step and construct feature vector for the concept nodes directly².

Semantic enrichment. The above step provides as output two ontologies, where documents have been assigned to each concept node in the ontologies. The next step is to calculate a feature vector for each concept node in the two ontologies respectively. To calculate a feature vector for each concept node, we first need to establish feature vectors for each document that belongs to the concept node.

Pre-processing. The first step is to transform documents, which typically are strings of characters, into a representation suitable for the task. The text transformation is of the following kind: remove HTML (or other) tags; remove stop words; perform word stemming (lemmatization). A linguistic workbench developed at Norwegian University of Science and Technology (NTNU) is employed [11].

Document representation. We use the vector space model [19] to construct the generalization of the documents. In vector space model, documents are represented by vectors of words. There are several ways of determining the weight of word i in a document d . We use the standard tf/idf weighting [19], which assigns the weight to word i in document d in proportion to the number of occurrences of the word in the document, and inverse proportion to the number of documents in the collection for which the word occurs.

Concept vector construction. We differentiate between leaf nodes and non-leaf nodes in the ontology. Leaf nodes are those which have no sub-nodes.

For each leaf node, the feature vector is calculated as an average vector on the documents vectors that have already been assigned to this node. Let C^k be the feature vector for concept node K and let D_j be the collection of documents

² An example, where documents have already been assigned to concept node is Open Directory Project <http://dmoz.org/>

that have been assigned to that node K . Then for each feature i of the concept node vector, it is calculated as:

$$C_i^k = \frac{\sum_{D_j \in K} w_{ij}}{|D_j|}$$

When it comes to non-leaf nodes, the feature vector C^k for a non-leaf concept node K is calculated by taking into consideration contributions from the documents that have been assigned to it, its direct sub nodes and the nodes with which concept K has relation³. Let D_j be the collection of documents that have been assigned to that node K , let S_t be the collection of its direct sub nodes and let S_r be the collection of its related nodes. The i th element of C^k is defined as:

$$C_i^k = \alpha * \frac{\sum_{D_j \in K} w_{ij}}{|D_j|} + \beta * \frac{\sum_{S_t \in K} w_{it}}{|S_t|} + \gamma * \frac{\sum_{S_r \in K} w_{ir}}{|S_r|}$$

where $\alpha + \beta + \gamma = 1$. α , β and γ are used as tuning parameters to control the contributions from the concept's instances, sub concepts, and related concepts respectively.

Mapping. The similarity of two concepts in two ontologies is directly calculated as the cosine measure between the two representative feature vectors. Let two feature vectors for concept a and b respectively, both of length n , be given. The cosine similarity between concept a and concept b is defined as:

$$\text{sim}(a, b) = \text{sim}(C^a, C^b) = \frac{C^a * C^b}{|C^a| * |C^b|} \hat{E} = \frac{\sum_{i=1}^n (C_i^a * C_i^b)}{\sqrt{\sum_{i=1}^n (C_i^a)^2} * \sqrt{\sum_{i=1}^n (C_i^b)^2}}$$

For concept a in ontology A , to find the most related concept b in ontology B , the top k ranked concept nodes in ontology B are selected according to the initial similarity measure calculated above. Those promising concepts are further evaluated by other matching strategies, which will be introduced in the post processing step. The similarity of relations is calculated based on the corresponding domain concepts and range concept of the relations. The similarity between relation $R(X, Y)$ and $R'(X', Y')$ is defined as :

$$\text{sim}(R, R') = \text{sim}(X, X') * \text{sim}(Y, Y')$$

$\text{sim}(X, X')$ and $\text{sim}(Y, Y')$ can be calculated by the above equation for concepts similarity.

³ At this point, all ad hoc relations other than subsumption are treated merely as related-to.

Post processing. Given the initial mapping suggested by the previous step, the users could choose to go for a post processing step to strengthen the prominent ones. WordNet [7] may be used to strengthen the mappings whose concept names have a close relatedness in WordNet.

In WordNet, nouns are organized into taxonomies where each node is a set of synonyms (a synset) representing a single sense. If a word has multiple senses, it will appear in multiple synsets at various locations in the taxonomy. These synsets contain bidirectional pointers to other synsets to express a variety of semantic relations. The semantic relation among synsets in WordNet that we use in this experiment is that of hyponymy/hypernymy, or the is-a-kind-of relation, which relates more general and more specific senses. Verbs are structured in a similar hierarchy with the relation being troponymy in stead of hypernymy.

One way to measure the semantic similarity between two words a and b is to measure the distance between them in WordNet. This can be done by finding the paths from each sense of a to each sense of b and then selecting the shortest such path. Note that path length is measured in nodes rather than links. So the length between sister nodes is 3. The length of the path between member of the same synset is 1. we did not make any effort in joining together the 11 different top nodes of the noun taxonomy. As a consequence, a path cannot always be found between two nouns. When that happens, the program returns a "not related" message.

The path length measurement above gives us a simple way of calculating relatedness between two words. However, there are two issues that need to be addressed.

- Multiple part-of-speech. The path length measurement can only compare words that have the same part-of-speech. It would be pointless to compare a noun and a verb, since they are located in different taxonomy trees. The words we compare in this context are concept names in the ontologies. Even though most of the names are made of single noun or noun phrase, verbs and adjectives do occasionally appear in a concept name label. In some cases, one word would have more than one part-of-speech (for instance, "backpacking" is both a noun and a verb in WordNet). For these words, we first check if it is a noun and if the answer is yes, we treat it as a noun. In the case of "backpacking", for instance, it will be treated as a noun and its verb sense will be disregarded. If it is not a noun, we check if the word is a verb and if the answer is yes, we treat it as a verb. Words that are neither nouns, verbs nor adjectives will be disregarded. This makes sense since the different part-of-speech of the same word are usually quite related and choosing one of them would be representative enough.
- Compound nouns. Those compound nouns which have an entry in WordNet (for example "jet lag", "travel agent" and "bed and breakfast") will be treated as single words. Others like "railroad transportation", which have no entry in WordNet, will be split into tokens ("railroad" and "transportation" in the example) and its relatedness to other word will be calculated as an average over the relatedness between each token and the other word. For

instance, the relatedness between "railroad transportation" and "train" will be the average relatedness of "railroad" with "train" and "transportation" with "train".

We have integrated the Java WordNet Library (JWNL) [5], a Java API, into the iMapper system for accessing the WordNet relational dictionary and calculate semantic relatedness based on the path length measurement described above. The computed relatedness will be amplified by a tuning parameter and then will be added to the similarity values computed in the previous step.

Presentation and Refinement. The iMapper system has been implemented with a graphical user interface. The users trigger each step by interacting with the GUI. The ontologies, and a list of top ranked mapping assertions are presented in the GUI. It is also possible for the user to edit, delete or add mapping assertions.

Translation and Storage. Once the approval and adjustment of mapping assertions are finished, the user can save the results. The mappings can be stored and exported in RDF (Resource Description Framework).

3 Evaluation

A comprehensive evaluation of the match processing strategies supported by the iMapper system has been performed on two domains. The main goal was to evaluate the matching accuracy of iMapper, to measure the relative contributions from the different components of the system, and to verify that iMapper can contribute to helping the user in performing the labor intensive mapping task.

3.1 Experiment Design

Performance criteria. To evaluate the quality of the match operations, we compare the match result returned by the automatic matching process (P) with manually determined match result (R). We determine the true positives, i.e. correctly identified matches (I). Based on the cardinalities of these sets, the following quality measures are computed.

$\text{Precision} = |I|/|P|$, is the fraction of the automatic discovered mappings which are correct. It estimates the reliability for the match prediction.

$\text{Recall} = |I|/|R|$, is the fraction of the correct matches (the set R) which have been discovered by the mapping process. It specifies the share of real matches that are found.

Domains and source ontologies. We evaluated iMapper on two domains. The product catalogue integration task was first introduced in [8], where consumers and vendors may use different classification schemas (UNSPSC, UCEC, and eCl@ss, to name a few) to identify their requests or products. Links between different classification schemas need to be defined in order to relate the corresponding concepts. In our experiment, the two relevant product catalogues are

the United Nations Standard Products and Services Code (UNSPSC)⁴ and the Standardized Material and Service Classificatione – eCl@ss⁵. UNSPSC contains about 20.000 categories organized into four levels. Each of the UNSPSC category definitions contains a category code and a short description of the product (for example, category 43191500 OPersonal communication devicesÓ). eCl@ss defines more than 12.000 categories and is organized in four-level taxonomy. It is generally understood that UNSPSC classifies products from a supplierÓs perspective, while eCl@ss is from a buyer’s perspective.

For our current experiment, two small segments of the relevant catalogues, both of which concern the domain of computer and telecommunication equipments, were selected. They contain 23 - 26 concepts (corresponding to the categories) and are organized in 3 - 4 levels by generalization relationships. Two datasets of product descriptions collected from online computer vendor websites are classified according to UNSPSC and eCl@ss. The classification is performed in two steps. First is the automatic classification by the CnS client, then come human adjustments of the automatic results. The classified product descriptions are viewed as the instances of the relevant concept.

The second domain we chose was the tourism section. The two ontologies are constructed based on vocabularies and structures from relevant travel sections of the Open Directory Project (ODP)⁶ and the Yahoo Category⁷. In both ODP and Yahoo, categories are organized in a hierarchy augmented with related-to links. In this domain, unlike the product catalogue example above, instances of each concept are already directly available without the need to classify them. For each category we downloaded the first 12 web site introductions. If there is less than 12 instances in the category, we downloaded all that is available.

Experiment setup. For the manual part, we conducted a user study in the Information System Group at the Norwegian University of Science and Technology. 7 users have conducted the manual mapping independently. All of them have good knowledge of modeling in general. None of the users has addressed the problem of mapping ontologies prior to the experiment. For each of the two mapping tasks, each participant received a package containing: a diagrammatic representation of the two ontologies to be matched, a brief instruction of the mapping task, and a scoring sheet to fill in the user identified mappings.

3.2 The Evaluation Result

Quality of iMapper’s predictions. For the two tasks, a number of mappings were identified manually by the users. From 7 results, one with highly deviating result was eliminated. Overall, an average of 30 mappings are discovered between the two product catalogues and an average of 62 in the tourism domain. The manual mappings are used as a gold standard to evaluate the quality of the

⁴ <http://www.unspsc.org>

⁵ <http://www.eclasse-online.com/>

⁶ <http://dmoz.org/>

⁷ <http://www.yahoo.com>

automatically suggested mappings. The automatic result is evaluated against each of the 6 manual mapping results to calculate the respective precision and recall. An average precision and an average recall are afterwards computed.

Figure 2(a) illustrates average precision versus recall figures for the two mapping tasks respectively. Since the mappings are ranked according to their similarity degrees so that the most similar ones are ranked high, the precision drops at lower recall levels. For the tourism ontology mapping task, the precision is 93% at recall level 10% and drops gradually when more mappings are included. For the product catalogue task, the precision at levels of recall higher than 70% drops to 0 because not all user identified mappings in this task can be discovered by the iMapper system automatically.

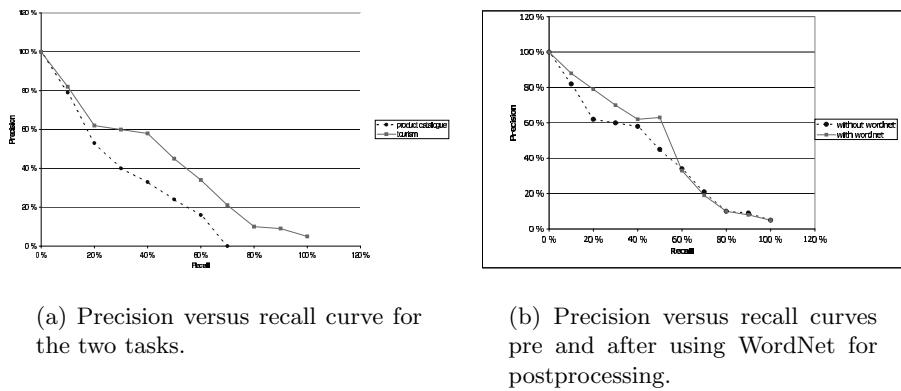


Fig. 2. Precision recall curves.

Note that the tourism ontology mapping task achieved higher precision than the product catalogue task at all recall levels. There are several possible explanations for this difference. First, the number of instances of the product catalogues is smaller than that of the tourism ontologies. As a result, the feature vectors generated by a larger instance set will have a better chance to capture and condense the terms that differentiate one concept from others. More accurate feature vectors will in turn boost the accuracy of mappings. Second, the significance of overlapping in content and structure of the to-be-mapped ontologies varies in the two tasks. It seems that the overlapping between the two tourism ontologies is larger than that between the two product ontologies. A higher overlapping makes it easier for the system to detect the mappings correctly. Third, the documents used in the two tasks have different characteristics. In the product domain, there exist a fair amount of technical terms, proper nouns and acronyms (for instance, "15inch", "Thinkpad", "LCD" etc.) in the product descriptions. Lacking of special means to treat these special terms hampers the system from

generating high quality feature vectors. In contrast, in the tourism domain, the documents contain far much less specific technical terms or proper nouns.

Further experiment. With the tourism domain, we did further experiment on assessing the effect of using WordNet[7] to post-process the system initially generated mappings. WordNet is used to strengthen the mappings whose concept names have a close relatedness in WordNet. In this experiment, the relatedness is defined as the hierarchical depth from one concept to the other in WordNet. Figure 2(b) shows the precision and recall curves pre and after using WordNet for post-processing in the tourism ontology mapping task. The figure shows that WordNet marginally improves the precision at levels of recall lower than 60%. This suggests WordNet is useful in strengthening the similarity value of the correct mappings and boost their ranks in the result set. The changing of ranks makes the predication more accurate at lower recall levels. We also observed the limitations of using WordNet to calculate concept relatedness. In WordNet, nouns are grouped by hyponymy/hypernymy relations into several hierarchies , each with a different unique beginner. Topic related semantic relations are absent in WordNet, so *travel agent* and *travel* have no relation between them. And in fact, they are in two different taxonomies, since *travel agent* is in the taxonomy which has *entity* as top node and *travel* is in the taxonomy where *act* is the top node. This results in a not-related result being returned when applying the path length measure for measuring the relatedness of the two terms. That result however does not mirror the human judgment. A possible way to overcome this limitation is to augment WordNet with domain specific term relations or use other domain specific lexicons.

Discussion. Even though the system discovered most of the mappings and ranked them in a useful manner, it is natural to ask how the system can achieve even better precision and recall figures. There are several reasons that prevent iMapper from correctly mapping the concepts. One reason is some of the questionable mappings the users identified. For example, one user mapped *Destination* with *Hitchhiking* and *Automotive* with *Railroad transportation*. On the other hand, there are also plausible mappings which the system discovered but no user has identified. For example, the system maps *Backpacking* with *Budget travel* but it has not been reported in any of the user's results. Further, the successful mapping is based on the successful construction of representative feature vectors which could differentiate one concept from another. The quality of the feature vectors is affected by the number of instance documents and the natural language processing techniques to extract textual information from the document in order to construct the feature vector. A solution to this problem is to utilize more document instances and employ more sophisticated textual processing techniques.

4 Related Work and Conclusions

There has been a number of works on semantic reconciliation developed in the context of schema translation and integration, knowledge representation, ma-

chine learning and information retrieval. In the multi database and information systems area, there exist approaches dealing with database schema integration. The work on database scheme integration is quite relevant, since schemas can be viewed as ontologies with restricted relationship types. DIKE [17], MOMIS [2], OBSERVER [15], and Cupid [12] are systems, which focus on schema matching. [10] has reported the usage of semantic enrichment for the comparison of dynamic aspect of schemas. In [18], a survey on automatic schema matching approaches was conducted.

In the research area of knowledge engineering, a number of ontology integration methods and tools exist. Among them, Chimaera [14] and PROMPT [16] are the few, which have working prototypes. [21] proposes a method called FCA-MERGE, based on the theory of formal concept analysis, for merging ontologies following a bottom up approach and the method is guided by application-specific instances of the given source ontologies that are to be merged. [1] uses techniques well known from the area of data mining (association rules) for the task of catalogue integration.

Our approach is in line with the latter group of endeavours. Yet it differs from them in the following ways. First information retrieval models are used to represent concept extension and calculate similarity between concepts. This gives us a practical way to rank the mapping results, which lead to the more prominent mappings be listed on top. When a large amount of mappings are predicated, ranking them is especially useful to help the user concentrate on the more likely to be correct ones. In addition, the leverage of vector space model allows taking into account the hierarchical information of the concept in a unified manner. Second, we use graphical notations to represent the ontology and the mapping results. This makes it easier for the user to understand the model and get an overview. Furthermore, we have explored the possibility of incorporating WordNet into the framework to achieve better mapping results.

The approach can be applied in several other different contexts, because of the domain independent nature of the approach. One such context is documents retrieval and publication between different web portals. Users may conform to their local ontologies through which the web portals are organized. It is desirable to have support for automated exchange of documents between the portals and still let the users keep their perspectives. Service matching is yet another candidate to apply the method, though we have to assume that there are some service description hierarchies (the MIT process handbook for instance [13]) and that the provider and the requester are using different classification schemas. By using the extension description of the service hierarchy, we can compute a feature vector for each service node. Then the matching can be conducted by calculating the distance between the representative feature vectors.

References

1. R. Agrawal and R. Srikant. On integrating catalogs. In *Proceeding of the WWW-11*, Hong Kong, 2001.

2. Sonia Bergamaschi, Silvana Castano, and Maurizio Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.
3. Tim Berners-Lee. The semantic web. *Scientific american*, 284(5):35–43, 2001.
4. Terje Brasethvik and Jon Atle Gulla. Natural language analysis for semantic document modeling. *Data & knowledge Engineering*, 38(1):45–62, 2001.
5. John Didion. Jwnl (java wordnet library), <http://sourceforge.net/projects/jwordnet/>, 2004.
6. DublinCore. <http://www.dublincore.org>.
7. Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
8. D. Fensel, Y. Ding, B. Omelayenko, E. Schulten, G. Botquin, M. Brown, and A. Flett. Product data integration in b2b e-commerce. *IEEE Intelligent Systems (Special Issue on Intelligent E-Business)*, 16(4):54–59, 2001.
9. Google, 2004.
10. Sari Hakkarainen. *Dynamic aspect and semantic enrichment in schema comparison*. PhD thesis, Stockholm University, 1999.
11. Harald Kaada. Linguistic workbench for document analysis and text data mining. Master's thesis, Norwegian University of Science and Technology, 2002.
12. J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching using cupid. In *Proceeding of Very Large Database Conference (VLDB) 2001*, 2001.
13. Thomas W. Malone, Kevin Crowston, Jintae Lee, and Brian Pentland. Tools for inventing organizations: Toward a handbook of organizational processes. Technical Report 141, MIT, 1993.
14. D. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning*, Colorado, USA, 2000.
15. E. Mena, A. Illarramendi, V. Kashyap, and A. P. Sheth. Observer: an approach for query processing in global infomation systems based on interoperation across pre-exist ontologies. *International journal on distributed and parallel databases*, 8(2):223–271, April 2000.
16. N. Fridman Noy and M. A. Musen. Prompt: algorithm and tool for automated ontology merging and alignment. In *Proceeding of American Association for Artificial Intelligence (AAAI) 2000*, 2000.
17. Luigi Palopoli, Giorgio Terracina, and Domenico Ursino. The system dike: Towards the semi-automatic synthesis of cooperative information systems and data warehouses. In *ADBIS-DASFAA Symposium 2000*, pages 108–117. Matfyz Press, 2000.
18. E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, 2001.
19. G. Salton and M. J. McGill. *An Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
20. Arne Solvberg. Data and what they refer to. In P. P. Chen, editor, *Concept Modeling: Historical Perspectives and Future Trends*. Springer Verlag, 1998.
21. G. Stumme and A. Maedche. Fca-merge: Bottom-up merging of ontologies. In *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI01.*, Seattle, USA, 2001.
22. Xiaomeng Su, Sari Hakkarainen, and Terje Brasethvik. Semantic enrichment for improving systems interoperability. In *Proceeding of the 2004 ACM Symposium on Applied Computing*, pages 1634–1641, Nicosia, Cyprus, 2004. ACM Press.
23. A. Valente, T. Russ, R. MacGregor, and W. Swartout. Building and (re)using an ontology for air campaign planning. *IEEE Intelligent Systems*, 14(1):27–36, 1999.

Testing Word Similarity: Language Independent Approach with Examples from Romance*

Mikhail Alexandrov¹, Xavier Blanco², and Pavel Makagonov³

¹Center for Computing Research, National Polytechnic Institute (IPN), Mexico
dyner@cic.ipn.mx, dyner1950@mail.ru

²Department of French and Romance Philology, Autonomous University of Barcelona
Xavier.Blanco@uab.es

³Mixteca University of Technology, Mexico
mpp2003@inbox.ru

Abstract. Identification of words with the same basic meaning (stemming) has important applications in Information Retrieval, first of all for constructing word frequency lists. Usual morphologically-based approaches (including the Porter stemmers) rely on language-dependent linguistic resources or knowledge, which causes problems when working with multilingual data and multi-thematic document collections. We suggest several empirical formulae with easy to adjust parameters and demonstrate how to construct such formulae for a given language using an inductive method of model self-organization. This method considers a set of models (formulae) of a given class and selects the best ones using training and test samples. We describe the method and give detailed examples for French, Italian, Portuguese, and Spanish. The formulae are examined on real domain-oriented document collections. Our approach can be easily applied to other European languages.

1 Introduction

Given a large text (or text corpus), we consider the task of grouping together the words having the same basic meaning, e.g., *sad*, *sadly*, *sadness*, *sadden*, *saddened*, etc.; this is usually referred as stemming. The algorithm can make two types of errors: to join words that are not similar (false positive) or fail to join words which are similar (false negative). We can tolerate a certain amount of such errors because our main motivation is to improve performance of information retrieval rather than to do some precise linguistic analysis.

To group together different letter strings they are often labeled in a specific way, the strings with the same label being considered pertaining to the same group. As labels, standard dictionary forms (i.e., singular for nouns, indefinite infinitive for verbs, etc.) can be used; reducing a string to such a form is called lemmatization. For lemmatization, morphology-based methods relying on morphological rules and a large morphological dictionary are usually used. They provide practically 100% accuracy on known words and good accuracy on the words absent in the dictionary [Gelbukh, 2003; Gelbukh and Sidorov, 2003, 2003].

* Work done under partial support of Mexican Government (CONACyT and CGEPI-IPN).

Alternatively, stemming can be used to label strings [Porter, 1980]: the words are truncated to their stems, which often reflect their invariant meaning; e.g., *sad*, *sadly*, *sadness*, *sadden*, *saddened* are truncated to the same stem *sad-*. Though this method is much simpler (since it usually relies only on lists of suffixes and suffix removal rules, but no dictionaries), it provides relatively low accuracy.

However, even such methods require large language-specific manually encoded rule sets with complicated structure. This becomes a problem in large-scale analysis of multilingual, multi-thematic document collections for which the effort required for manual compilation of suffix removal rules is prohibitive. In this paper we investigate the possibilities of using simple formulae with automatically trainable parameters.

Makagonov and Alexandrov [2002] have suggested an empirical method for testing word similarity and described a methodology for constructing empirical formulae based on the number of the coincident letters in the initial parts of the two words and the number of non-coincident letters in their final parts. The limited experiments on English texts with one of the formulae showed that for the documents from a specific domain it provides accuracy of 85% to 92%. The main advantage of this knowledge-poor approach is its simplicity and flexibility. It does not rely on any manually compiled morphological rules, dictionaries, suffix lists, or rule systems. To adapt the method to a new language or a new subject domain, only the parameters of the formula are to be automatically adjusted.

Still many important issues concerning application of the empirical formulae remain open. In this paper we give more detail on this method, investigate the sensitivity of the formulae to different languages, and analyze the errors the method commits. In particular, we show that, similarly to other statistical characteristics [Gelbukh and Sidorov, 2001], the parameters of such formulae depend on language.

We considered here the Romance languages: French, Italian, Portuguese, and Spanish, in the domain widely discussed currently in the European Community: mortgage and crediting. To evaluate the accuracy of the results, we use characteristics both from mathematical statistics and from information retrieval.

2 The Problem

2.1 Formulae for Testing Word Similarity

Our empirical formulae to be constructed test some hypothesis about word similarity. We consider only the languages where the word's base (the morphologically invariant part) is located at the beginning of the word (and not, say, at the end). It is generally true for the European languages.

Thus the formula relies on the following characteristics of the pair of words:

- n*: the total number of *final* letters differing in the two words,
- y*: the length of the common *initial* substring,
- s*: the total number of letters in the two words,

so that $2y + n = s$. For example, for the words *sadly* and *sadness*, the maximal common initial substring is *sad-*, thus the differing final parts are *-ly* and *-ness* so that $n = 6$, $y = 3$, and $s = 12$.

We consider each formula as a model from a given class. The models differ in the number of parameters, which define the model complexity. Our problem is to select the optimal complexity of the model.

In this paper we will consider the following class of models for making decisions about word similarity: Two words are similar if and only if the relative number of their differing final letters is less than some threshold depending on the number of initial coincident letters of the two words:

$$\frac{n}{s} \leq F(y), \quad F(y) = a + b_1y + b_2y^2 + \dots + b_ky^k, \quad (1)$$

where n , s , and y are as defined above, $F(y)$ is the model function. Such function presentation is general enough because any continuous function can be represented as a convergent polynomial series.

Obviously, such models have two degrees of liberty, n and y , with respect to the characteristics of the word pair. One can also consider a model in the form $n/s \leq F(y/s)$ with only one degree of liberty since $y/s = (1 - n/s)/2$; however, the form (1) is more flexible, as our experiments clearly prove.

Note that more general classes of models can be suggested, as, for example, the following one: Two words are similar if the distance between them satisfies the inequality

$$D(y, s_1, s_2) \leq 0, \quad D(y, s_1, s_2) = \sum F(t) * \delta(t), \quad t = (y+1), \dots, N \quad (2)$$

where s_1 and s_2 are the lengths of the words, N is accepted maximum number of letters in words of a given language, (t) is an indicator, $F(t)$ is a model (penalty) function. Here $(t) = \{0, 0.5, 1\}$ if there are zero, one or two letters in a correspondent position t of both words, and $F(t)$ can be represented in a polynomial form, inverse polynomial form, or in any other form. However, in this paper we concentrate on the models in the form (1).

Our first task is to find the best form of model function, i.e., for the models (1), the degree of the polynomial. Then, after the model has been selected, it is easy to determine its optimal parameters.

2.2 Limitations of the Approach

First, our approach for testing word similarity is not applicable to irregular words. Indeed, it is impossible to construct a simple formula that could detect any similarity between English irregular verbs such as *buy* and *bought*, because these words have only one common letter in the initial part of the string. The same situation occurs in the Romance languages considered in this paper.

Since the empirical formula is constructed on the basis of statistical regularities of a language, it leads to the above-mentioned errors of the first and the second kinds (false positive and false negative). Tuning the model parameters we can control the balance between these two kinds of errors, but not to completely avoid the errors.

Some specific errors can be caused by fuzzy sense of some base meanings; we call such kind of errors the errors of 3-rd type. As an example, consider the words *move*, *moving*, *moveable*, and *moveability*. The latter two words can be considered either as

having the same basic meaning as the former two or as differing from them in the additional meaning ‘ability’. Any extended interpretation of base meanings by the user leads to constructing a formula with a high level of errors of the first kind. In our example it is better to consider the similarity between the words *move* and *moving* reflecting the base meaning *movement*, and between *moveable* and *moveability* reflecting the basic meaning *ability to movement*. In this case, the formula has significantly lower error level. Obviously the errors of the 3-rd kind are more typical for document sets in special domains [Porter, 1980]. For example, the words *relate* and *relativity* can be considered similar in general texts but not in texts on physics.

3 Inductive Method of Model Self-Organization

3.1 Contents of the Method

The inductive method of model self-organization (IMMSO) [Ivahnenko, 1980] allows determining a model of optimal complexity for various processes or phenomena. The method chooses a near-optimal model in a given class of models using experimental data. It cannot find the very optimal model in a continuous class because it is based on competition of models; this is why the method is called *inductive*.

The method consists in the following steps:

- (1) An expert defines a sequence of models, from simplest to more complex ones.
- (2) Experimental data are divided into two data sets: training data and test data, either manually or using an automatic procedure.
- (3) For a given kind of model, the best parameters are determined using the training data and then using the test data. Here any internal criteria¹ of concordance between the model and the data may be used, e.g., the least squares criterion.
- (4) Both models are compared on the basis of external criteria (see below), such as the criterions of regularity and unbiasedness.
- (5) The external criteria (or the most important one) are checked on having reached a stable optimum. In this case the search is finished. Otherwise, more complex model is considered and the process is repeated from the step 3.

Here is why the external criteria reach an optimum (minimum). The experimental data is supposed to contain: (a) a regular component defined by the model structure and (b) a random component—noise. Obviously, the model is to be capable to reflect the changes of the regular component. When the model is too simple, it is weakly sensible to this regular component and insensible to the noise. When the model is too complex, it reflects well the regular component but also the changing of the random component. In both cases the values of the penalty function (criterion) are large. So, we expect to find a point where the criterion reaches its minimum. The principle of model *auto-organization* consists in that an external criterion passes its minimum when the complexity of the model is gradually increased.

¹ Internal criteria use the same data for both evaluation of model quality and defining its parameters. External criteria use different data for these purposes. Usually the external criteria are constructed as non-negative functions with the best value 0.

3.2 Application of the Method

In order to apply IMMSO to the problem of construction of the formulae, we consider the extreme cases of the equation (1), i.e., examples of word pairs for which (1) becomes an equality. This gives a system of linear equations with respect to unknown parameters a, b_1, b_2, \dots, b_k :

$$n_i/s_i = a + b_1 y_i + b_2 y_i^2 + \dots + b_k y_i^k, \quad i = 1, \dots, m \quad (3)$$

Here n , s , and y are defined as in Section 2.1, and i is the number of examples prepared by an expert on the given language. Of course, the number of equations m should be more than the number of variables ($k + 1$). To filter out the noise, the number of equations should be at least 3 times greater than the number of variables.

The examples forming the system (3) must be prepared by an expert on the given language and can be considered experimental data. Consider, for example, the words *hoping* and *hopefully*. These two words have very short common part *hop-* and long different parts *-ing* and *-efully*. The corresponding equations are:

$$\begin{aligned} 9/15 &= a + 3b_1 && \text{for linear model,} \\ 9/15 &= a + 3b_1 + 9b_2 && \text{for quadratic model, etc.} \end{aligned}$$

The next steps according to IMMSO methodology are: for several k starting with $k = 1$, the best solution of (4) for a given internal criterion is found and then the external criterion(s) are checked on having reached the minimum.

3.3 External Criteria

Generally IMMSO uses the following two criteria:

- criterion of regularity
- criterion of unbiasedness

Both criteria use the training data set and the test data set. The criterion of regularity reflects the difference between the model and the testing data, while the model is constructed on the training data set. So, this criterion evaluates the stability of the model with respect to data variation. The criterion of unbiasedness reflects the difference between the two models—those constructed on the training and on the testing set, respectively. So, this criterion evaluates independence of the model from the data.

Different forms of these criteria can be proposed, a specific form depending on the problem. In our case we use these criteria in the following forms:

$$K_r = \frac{\sqrt{\sum_c (q_i(T) - q_i)^2}}{\sqrt{\sum_c (q_i)^2}} \quad K_u = \frac{\sqrt{\sum_{T+C} (q_i(T) - q_i(C))^2}}{\sqrt{\sum_{T+C} (q_i)^2}} \quad (4)$$

Here T and C are the systems of equations (3) used for training and testing, respectively; $q_i(T)$ and $q_i(C)$ are the “model” data that is the right part of equations with the parameters determined on the data of training and testing, respectively; q_i are the “experimental” data, i.e., the left part of the equations; i is the number of the equation.

Sometimes a model can be better than another one according to the first criterion but worse according to the second one. Then a combined criterion is used, e.g.:

$$K = \lambda K_r + (1-\lambda) K_u, \quad (5)$$

where λ is a user-defined coefficient of preference. In our experiments with Romance languages we use $\lambda = 2/3$, i.e., we consider the criterion of regularity as the main one.

4 Constructing Empirical Formula for Romance Languages

4.1 Selection of Examples

The formula to be constructed is considered as the first approximation and may be later tuned on the texts from a given domain. So, our examples selected for training and control reflect only some general regularities of a language.

We assumed that: (a) the initial common parts of similar words were distributed uniformly between the shortest and the longest ones; (b) the final parts of similar words were also distributed uniformly between the shortest and the longest ones. So, we took 50% of word pairs with the shortest common part and 50% with the longest common part. In each pair, we tried to take the words with the short and long final parts. We did not consider the words containing less than 4 letters and we removed diacritics from letters (i.e., ê, é, è → e, ç → c).

The number of examples taken for training and testing was 10, which corresponded to the expected maximum number of model parameters of 4 (cubic polynomial).

4.2 Training Procedure

For training procedure, we considered the following pairs of words having the same basic meanings:

French

N	Short words		N	Long words	
1.	Blanc	<i>Blancheur</i>	6.	<i>Impossible</i>	<i>Impossibilite</i>
2.	Pleurant	<i>Pleurerait</i>	7.	<i>Degenerer</i>	<i>Degenerescent</i>
3.	Mangeur	<i>Mangerent</i>	8.	<i>Macadam</i>	<i>Macadamiser</i>
4.	Guet	<i>Gueteurs</i>	9.	<i>Pauvrete</i>	<i>Pauvrement</i>
5.	Blessant	<i>Blessures</i>	10.	<i>Abrutissant</i>	<i>Abrutissement</i>

Italian

N	Short words	N	Long words		
1.	Sebo	Seborrea	6.	Convertire	Convertibile
2.	Bello	Belta	7.	Convinzione	Convincimento
3.	Arte	Artistico	8.	Fiammifero	Fiammeggiare
4.	Casa	Casigliano	9.	Macchinatore	Macchinazione
5.	Alter	Alterarsi	10.	Panellenico	Panellenismo

Portuguese

N	Short words	N	Long words		
1.	Dossel	Dosseladas	6.	Abandonando	Abandonamento
2.	Dotar	Dotacaos	7.	Amoravel	Amoravelmente
3.	Hipnose	Hipnotico	8.	Celebridades	Celebrizaram
4.	Janta	Jantarada	9.	Catalogacao	Catalogadora
5.	Jardim	Jardineiro	10.	Caracteri- zante	Caracterizador

Spanish

N	Short words	N	Long words		
1.	Celo	Celosamente	6.	Arrogante	Arrogancia
2.	Cazar	Cazador	7.	Institucional	Institucionalmente
3.	Arte	Artistico	8.	Multiplicados	Multiplicaciones
4.	Comer	Comida	9.	Descentralizados	Descentralizables
5.	Altura	Altitud	10.	Caracteristica	Caracterizaremos

With these examples, we obtained the following systems of linear equations for French:

N	0 th order model	N	Linear model	N	Quadratic model
1.	$4/14 = a$	1.	$4/14 = a + 5b_1$	1.	$4/14 = a + 5b_1 + 25b_2$
2.	$8/18 = a$	2.	$8/18 = a + 5b_1$	2.	$8/18 = a + 5b_1 + 25b_2$
3.	$6/16 = a$	3.	$6/16 = a + 5b_1$	3.	$6/16 = a + 5b_1 + 25b_2$
4.	$5/13 = a$	4.	$5/13 = a + 4b_1$	4.	$5/13 = a + 4b_1 + 16b_2$
5.	$7/17 = a$	5.	$7/17 = a + 5b_1$	5.	$7/17 = a + 5b_1 + 25b_2$
6.	$7/23 = a$	6.	$7/23 = a + 8b_1$	6.	$7/23 = a + 8b_1 + 64b_2$
7.	$6/22 = a$	7.	$6/22 = a + 8b_1$	7.	$6/22 = a + 8b_1 + 64b_2$
8.	$4/18 = a$	8.	$4/18 = a + 7b_1$	8.	$4/18 = a + 7b_1 + 49b_2$
9.	$6/18 = a$	9.	$6/18 = a + 6b_1$	9.	$6/18 = a + 6b_1 + 36b_2$
10.	$8/24 = a$	10.	$8/24 = a + 8b_1$	10.	$8/24 = a + 8b_1 + 64b_2$

Similar linear system was also constructed for the cubic model. Using the least squares method, we found four sets of model parameters (a, b_1, \dots) for each of the mentioned models, to be further used in the external criterions.

The models for Italian, Portuguese, and Spanish were constructed in a similar way.

4.3 Testing Procedure

For testing procedure, we considered the following pairs of words having the same base meanings:

French

N	Short words	N	Long words
1.	Froid Froideur	6.	Eminent Eminemment
2.	Mort Mortelle	7.	Epigramme Epigrammatique
3.	Ferais Ferrer	8.	Retentissant Retentissement
4.	Fin Finalite	9.	Constitutionnel Constitutionnalisme
5.	Lutter Luttait	10.	Difficulte Difficultueux

Italian

Short words		Long words	
1.	Certo Certezza	6.	Perforatora Perforazione
2.	Forca Forchetta	7.	Periodico Periodizzare
3.	Mostro Mostruosita	8.	Cattedra Cattedratico
4.	Pane Panettone	9.	Cattechismo Cattechizzatore
5.	Balle Ballerina	10.	Necessaria Necessariamente

Portuguese

Short words		Long words	
1.	Pequeneza Pequeninas	6.	Descompassar Descompassadamente
2.	Dourar Douradura	7.	Experimentou Experimentavel
3.	Macho Machista	8.	Impugnar Impugnativo
4.	Nodo Nodosidade	9.	Doutoral Doutoramento
5.	Nome Nominalidade	10.	Necessitadas Necessidades

Spanish

Short words		Long words	
1.	Circo Circense	6.	Sentimentales Sentimentalismo
2.	Afan Afanoso	7.	Discriminamos Discriminacion
3.	Denso Densidad	8.	Legislador Legislatura
4.	Caber Cabida	9.	Especialista Especializarse
5.	Creado Creacion	10.	Necesario Necesariamente

On the basis of these examples we constructed the correspondent systems of linear equations for the four models like we did in the training procedure. Using the least squares method, we found again the four sets of the model parameters (a, b_1, \dots) for each of the mentioned models. Of course, these models differed from the models constructed on the testing set.

4.4 Results

Using the criteria (4) and (5), we found that the linear model proved to be the winner for all languages. The results are summarized in the following tables:

French

	0-order model	Linear model	Quadratic model	Cubic model
Criterion K_r	0.28	0.20	0.29	5.27
Criterion K_u	0.01	0.04	0.18	3.81
Criterion K	0.19	0.15	0.25	4.78

Italian

	0-order model	Linear model	Quadratic model	Cubic model
Criterion K_r	0.28	0.20	0.21	0.33
Criterion K_u	0.15	0.10	0.14	0.20
Criterion K	0.24	0.17	0.19	0.29

Portuguese

	0-order model	Linear model	Quadratic model	Cubic model
Criterion K_r	0.31	0.23	0.25	2.25
Criterion K_u	0.13	0.14	0.17	2.41
Criterion K	0.25	0.20	0.22	2.30

Spanish

	0-order model	Linear model	Quadratic model	Cubic model
Criterion K_r	0.26	0.19	0.18	0.23
Criterion K_u	0.09	0.11	0.12	0.18
Criterion K	0.20	0.16	0.16	0.21

At the last step for all languages we considered only the linear model. We joined together the training set and the testing set of examples and obtained the linear system of 20 equations and 2 variables. The solutions gave the following formulae for testing word similarity:

French:	$n/s \leq 0.481 - 0.024 y$
Italian:	$n/s \leq 0.571 - 0.035 y$
Portuguese:	$n/s \leq 0.528 - 0.029 y$
Spanish:	$n/s \leq 0.549 - 0.029 y$

Similarly, joining together all 80 examples we determined the generalized formula:

$$n / s \leq 0.530 - 0.029 y.$$

This formula can be considered an initial approximation for further tuning on other romance languages.

5 Experimental Results

5.1 Document Collections

The constructed formulae were checked on real document collections. The goals of these experiments were:

- To compare the quality of the formulae on different languages,
- To reveal the sensibility of each formula to its parameters.

We considered the documents on the popular theme: mortgage and crediting. This theme is narrow enough to provide a representative set of similar words. We took 6 articles in each language containing in total from 16000 to 24000 words (excluding numbers and words with less than 4 letters). The statistical characteristics of style for each document set were rather close, namely: 22.6–24.4 for text complexity and 0.14–0.17 for word's resource variety. The first figure is calculated with as $n * \ln m$ and the second one as N / M , where n and m are the average length of words (in letters) and phrases (in words), N and M are the number of different words (before grouping) and that of all words. Therefore, the conditions for all languages were the same.

To reduce the number of comparisons, we randomly selected some paragraphs from the mentioned document collections. Since we assumed that similar words had different final parts, it was natural to order all words alphabetically and to compare the neighbors (this is not necessarily the best way for grouping similar words, but we used it to simplify manual testing of word pairs). As a result we obtained the alphabetical lists of words partially presented in the following table:

French	Italian	Portuguese	Spanish
absence	accedere	abrange	abaratado
accepte	acquistare	abrangencia	abogado
accord	acquisto	abrangendo	abogados
accorde	adatte	acessoes	acceder
accordee	adatto	acima	actual
...etc...	...etc...	...etc...	...etc...
In total: 456 words	In total: 567 words	In total: 506 words	In total: 536 words

5.2 Results

In our experiments we tested similarity of adjacent words automatically and manually with different combinations of parameters used in the formulae, varied in $\pm 10\%$ for each parameter. The quality of the results was estimated by the total statistical error

$P_{\text{err}} = P_p + P_n$ and F-measure of accuracy $F = 2 / (1/R + 1/P)$. Here P_p and P_n are the probabilities of the statistical errors of the 1-st and the 2-nd kind; R and P are recall and precision. The first estimation is usually used in mathematical statistics [Cramer, 1946], while the second one in information retrieval [Baeza-Yates, 1999].

The following tables show the result of automatic processing for different languages.

French (455 tests = 115 similar + 340 non-similar)

Parameters	0.48, -0.024	0.43, -0.024	0.53, -0.024	0.48, -0.021	0.48, -0.027
Similar cases	104	88	126	111	100
Not similar	351	367	329	344	355
False alarm	7	3	18	10	7
Omission	18	30	7	14	22
False positive P_p	2.0%	0.9%	5.3%	2.9%	2.0%
False negative P_n	15.7%	26.1%	6.1%	12.2%	19.1
Recall R	84.3%	73.9%	93.9%	87.8%	80.9%
Precision P	93.3%	96.6%	85.7%	91.0%	93.0%
Summary	Min $P_{\text{err}} = 11.4\%$		Max F = 89.7%		

Italian (566 tests = 140 similar + 426 non-similar)

Parameters	0.57, -0.035	0.51, -0.035	0.63, -0.035	0.57, -0.031	0.57, -0.039
Similar cases	149	120	193	166	126
Not similar	417	446	373	400	440
False alarm	39	19	65	45	19
Omission	30	39	12	19	33
False positive P_p	9.2%	4.2%	15.3%	10.6%	4.5%
False negative P_n	21.4%	27.9%	8.6%	13.6%	23.6%
Recall R	78.6%	72.1%	91.4%	86.4%	76.4%
Precision P	73.8%	84.2%	66.8%	72.9%	84.9%
Summary	Min $P_{\text{err}} = 23.9\%$		Max F = 80.3%		

Portuguese (505 tests = 138 similar + 367 non-similar)

Parameters	0.53, -0.029	0.48, -0.029	0.58, -0.029	0.53, -0.026	0.53, -0.032
Similar cases	136	117	159	141	132
Not similar	369	388	346	364	373
False alarm	15	9	27	17	14
Omission	17	30	6	14	20
False positive P_p	4.1%	2.5%	7.4%	4.6%	3.8%
False negative P_n	12.3%	21.7%	4.3%	10.1%	14.5%
Recall R	87.7%	78.3%	95.7%	89.9%	85.5%
Precision P	89.0%	92.3%	83.0%	87.9%	89.4%
Summary	Min $P_{\text{err}} = 11.7\%$		Max F = 89.3%		

Spanish (535 tests = 165 similar + 370 non-similar)

Parameters	0.55, -0.029	0.49, -0.029	0.61, -0.029	0.53, -0.026	0.53, -0.032
Similar cases	164	128	183	167	142
Not similar	371	407	352	368	393
False alarm	22	8	34	23	14
Omission	23	45	16	21	37
False positive P_p	5.9%	2.2%	9.2%	6.2%	3.8%
False negative P_n	13.9%	27.3%	9.7%	12.7%	22.4%
Recall R	86.1%	72.7%	90.3%	87.3%	77.6%
Precision P	86.6%	93.8%	81.4%	86.2%	90.1%
Summary	Min $P_{err} = 18.9\%$		Max F = 86.6%		

Examples of errors of all kinds are presented at the following table:

Errors	French	Italian	Portuguese	Spanish
1 st kind	commune	casa	entre	ahora
	communiqué	caso	entrega	ahorro
2 nd kind	hypotheque	iniziali	entendemos	invertido
	hypothecaire	inizialmente	entendimiento	inversores
3 rd kind	simple	numero	titulo	bancarios
	simplifie	numeroso	titulares	bancarrota

6 Conclusions

We have suggested a knowledge-poor approach for testing word similarity. Our empirical formulae do not require any morphological dictionaries of the given language and can be constructed manually or automatically basing on few examples. This is useful for constructing word frequency lists when working with multilingual databases and multi-thematic document collections.

Our experiments with Romance languages show that our approach provides the 80%–90% accuracy (F-measure), committing 2%-5% of the errors of the 1-st kind and 20%-25% of the 2-nd kind. This is rather acceptable in semi-automatic setting since the human expert can easily join the similar words after the grouping procedure.

In the future we plan to construct several other empirical formulae and compare them with those reported in this paper. We plan to give linguistic explanation for the behaviour of constructed formulae if possible. We plan also to take into account some statistical regularities extracted from the training document set. We thank A. Gelbukh and M. Porter for useful suggestions on such modifications.

References

1. Baeza-Yates, R., Ribero-Neto, B. (1999): *Modern Information Retrieval*. Addison Wesley.
2. Cramer, H. (1946): *Mathematical methods of statistics*. Cambridge.
3. Gelbukh, A. (2003): Exact and approximate prefix search under access locality requirements for morphological analysis and spelling correction. *Computación y Sistemas*, vol. 6, N 3, 2003, pp. 167–182.

4. Gelbukh, A., G. Sidorov (2001): Zipf and Heaps Laws' Coefficients Depend on Language. *Computational Linguistics and Intelligent Text Processing* (CICLing-2001). Lecture Notes in Computer Science, N 2004, Springer, pp. 332–335.
5. Gelbukh, A., G. Sidorov (2002): Morphological Analysis of Inflective Languages through Generation. *Procesamiento de Lenguaje Natural*, No 29, 2002, p. 105–112.
6. Gelbukh, A., G. Sidorov (2003): Approach to construction of automatic morphological analysis systems for inflective languages with little effort. In: *Computational Linguistics and Intelligent Text Processing* (CICLing-2003), Lecture Notes in Computer Science N 2588, Springer, pp. 215–220.
7. Ivahnenko, A. (1980): *Manual on typical algorithms of modeling*. Tehnika Publ., Kiev (in Russian).
8. Makagonov, P., M. Alexandrov (2002): *Constructing empirical formulas for testing word similarity by the inductive method of model self-organization*. In: Ranchhold and Mamede (Eds.) “Advances in Natural Language Processing”, Springer, LNAI, N 2379, pp. 239–247
9. Porter, M. (1980): An algorithm for suffix stripping. *Program*, 14, pp. 130–137.

Language Modeling for Effective Construction of Domain Specific Thesauri

Libo Chen and Ulrich Thiel

Fraunhofer IPSI
Dolivostr.15
64293 Darmstadt, Germany
+49 (0)6151 869 957
`{chen, thiel}@ipsi.fraunhofer.de`

Abstract. In this paper we present an approach for effective construction of domain specific thesauri. We assume that the collection is partitioned into document categories. By taking advantage of these pre-defined categories, we are able to conceptualize a new topical language model to weight term topicality more accurately. With the help of information theory, interesting relationships among thesaurus elements are discovered deductively. Based on the “Layer-Seeds” clustering algorithm, topical terms from documents in a certain category will be organized according to their relationships in a tree-like hierarchical structure --- a thesaurus. Experimental results show that the thesaurus contains satisfactory structures, although it differs to some extent from a manually created thesaurus. A first evaluation of the thesaurus in a query expansion task yields evidence that an increase of recall can be achieved without loss of precision.

1 Introduction

Many attempts to cope with the contemporary explosive growth of information available from various sources are based on more or less systematic approaches to organizing the knowledge about given domains in ontologies or thesauri. Topical term structures, especially hierarchical topical term structures like thesauri, are an effective way to represent knowledge about a specific domain. By utilizing the appropriate thesauri, a user can quickly gain an overview of a knowledge area and easily access the proper information. Such thesauri can be first of all utilized for query refinement. It is well known that users are not always able to formulate proper queries for search engines. The queries are either too short or the terms in the queries sometimes differ from those in search engines’ indices. These problems normally arise when users lack the necessary domain knowledge or they have to use a foreign language. Both query deficiencies lead to unsatisfactory information retrieval results. As thesauri are a good summary of specific knowledge domains, users can apply them to refine their original queries. The refinement can be done either automatically or interactively. On the search engine side, in order to minimize the gap between vocabularies of users and that of search engines, thesauri can serve as a tool to aid document indexing by providing standardised domain specific terms. In addition, thesauri can be applied as pre-constructed frames for ontology construction and document categorization.

Although there already exist a number of approaches for automatic construction of term structures in a certain document set, few of them can be directly put into practice, because fully automatic processing based on the “raw” data from unorganized resource documents often leads to expensive computational processing and unconvincing results. Manually made domain ontologies and thesauri possess much better structures for representing domain knowledge. However, constructing and maintaining such structures prove to be extremely resource consuming and inflexible.

We note that numerous pre-defined category systems ranging from big web directories such as Yahoo! to digital library categories in universities exist, where documents are classified into previously defined categories by human experts. We presumed that by exploiting such categories which reflect human intelligence, we are able to gain a much better view upon the documents and the terms contained in those documents. Thus, more reasonable words could be chosen for thesaurus construction and more effective clustering algorithms for term organization could be developed.

In our work, we present an approach for constructing a thesaurus automatically by taking advantage of pre-defined categories. A new language model is developed for defining term topicality. New interesting relationships among thesaurus elements are discovered by deeply analysing the usage of the pointwise mutual information measure for term association calculation. A clustering method --- the Layer-Seeds Method is applied to organize terms in a thesaurus that features a tree-like hierarchical structure. First step experiments show positive results both in the thesaurus structure and in the application of the thesaurus for automatic query expansion.

In the next section we review some existing techniques, which are related to our work. In section 3 and 4 we discuss how to conceptualize a topical language model for calculating term topicality and show the determination of relationships among thesaurus elements with help of information theory. In section 5 the Layer-Seeds clustering algorithm is introduced. In section 6 we discuss the results of first evaluations. The last section draws conclusions and points to future work.

2 Related Work

Due to their importance, thesauri have been the focus of research for a long time. Previous work in thesaurus construction can be roughly divided into two groups: manual construction and automatic construction.

Manual thesauri include general-purpose thesauri like Roget’s and WordNet, which contain sense relations like antonym and synonym. Another type of manual thesauri, which are used more specifically, usually contain relations between thesaurus items such as BT (Broader Term), NT (Narrow Term), UF (Use For) and RT (Related To). They include, for instance, INSPEC, LCSH (Library of Congress Subject Headings) and MeSH (Medical Subject Headings). The major problem with manual thesauri is the high cost and inflexibility in their construction and maintenance.

Research related to automatic thesaurus construction dates back to Sparck-Jones’s work on automatic term classification [9] and Salton’s work on automatic thesaurus construction and query expansion [7]. Later research includes the work of Crouch & Yang [1], Qiu & Frei [6] and Jing & Croft [4]. Sanderson et. al. [8] proposed an ap-

proach to build subsumption hierarchies. Term topicality is guaranteed by selecting terms from top-ranked documents retrieved for a query. A term is accepted as a sub-topic if $P(\text{topic}|\text{subtopic}) \geq 0.8$, where the value of 0.8 was chosen through informal analysis of subsumption term pairs. Lawrie [5] was able to create hierarchical summaries for document collections. She employed KL divergence to determine term topicality and a co-occurrence language model was used to calculate the probability of predictiveness for terms in different levels.

By applying pre-defined category systems, we are able to combine the advantages of both manually and automatically constructed thesauri, i.e. accuracy, low cost and flexibility. Furthermore, while Sanderson and Lawrie only established term relations between different layers of a hierarchy, two kinds of term relationships can be determined in our thesaurus: the “Thematical Specialization” (TS)- and the “Mutually Relevant” (MR)-relationship. The TS-relationship, discovered by theoretical deduction, can be seen as a more adaptable expansion of Sanderson’s subsumption-relationship and the MR-relationship shows the relation among terms in the same layer of a hierarchy.

3 Term Weighting – Calculating the Term Topicality

The weight of a term can be given either according to its occurrence or to its topicality. Traditionally, the importance of a term is judged solely by its occurrence for documents, i.e. how well a term is able to distinguish some documents from others according to its occurrences in a document collection. A typical example for it is the tf-idf weighting scheme. Recent research [8, 5] shows, however, that for a certain topic, term topicality is much more suitable for term weighting than term occurrence, since it specifically measures how well the term is able to represent a topic. The topicality of a term is normally calculated by comparing its distribution in a certain document collection representing a special topic to its distribution in general language or the whole document collection [8, 5].

In a pre-defined category system, which is the basis for our work, the calculation of both occurrence and topicality are different than that in an unorganized document collection in previous research. It is interesting to find out how these two measures can be calculated and which one is more suitable for term weighting in a category system. In the following we first introduce a notation system to describe a category system formally. Then we take a closer look at how the weight of a term in a category system can be given based on its occurrence. Derived from this analysis a topical language model will be finally deduced to calculate the weight of a term according to its topicality for a category.

3.1 Formal Definition of Pre-defined Category System

A category system C can be seen as a set of pre-defined categories C_i , which represent different special topics. C will then be denoted as $\{C_i, i=1,2,\dots,n1\}$. Viewed as “a huge bag of words”, C can also be described as a set of terms $\{w_t, t=1,2,\dots,n2\}$. A category C_i can be described in a similar way. As a bag of words, C_i can be directly

denoted as $\{w_t^{C_i}, t=1,2,\dots,n3\}$, where $w_t^{C_i}$ denotes the term w_t in category C_i . If we take the documents in C_i into consideration, then C_i consists of a set of documents $\{D_j^{C_i}, j=1,2,\dots,n4\}$, with $D_j^{C_i}$ as the document D_j in category C_i . A document $D_j^{C_i}$ further consists of a set of words $\{w_t^{D_j^{C_i}}, t=1,2,\dots,n5\}$, where $w_t^{D_j^{C_i}}$ denotes the term w_t in document $D_j^{C_i}$ in category C_i . In our work, thesauri are always constructed with respect to a certain category C_i . The occurrence and topicality of a term also refers to the corresponding category.

3.2 Occurrence-Based Term Weighting in a Pre-defined Category

In a category system, term occurrence can be defined as the contribution of the occurrence of a term to the occurrence of a category. $P(C_i|w_t)$ is a reasonably good measure for calculating the term occurrence of w_t to C_i . It provides information on how possible a category C_i will appear with the condition that w_t appears. According to Bayes' Theorem we have

$$P(C_i|w_t) = \frac{P(w_t | C_i) \times P(C_i)}{P(w_t)}. \quad (3.1)$$

To a certain C_i , $P(C_i)$ remains constant for all w_t . Therefore, it will not be dealt with in the following discussion. The two remaining factors in this formula are $P(w_t|C_i)$ and $P(w_t)$, where $P(w_t|C_i)$ provides a positive and $P(w_t)$ a negative contribution to the value of $P(C_i|w_t)$.

In case that C_i is already given, $P(w_t|C_i)=P(w_t^{C_i})=P(w_t^{C_i} \cap C_i)$. A traditional method to estimate $P(w_t^{C_i})$ in C_i is using a unigram language model, where C_i is considered as "a bag of words". $P(w_t^{C_i})$ is then calculated as follows:

$$P(w_t^{C_i}) = \left| w_t^{C_i} \right| / \left| w^{C_i} \right|. \quad (3.2)$$

where $\left| w_t^{C_i} \right|$ denotes the frequency of $w_t^{C_i}$ in C_i and $\left| w^{C_i} \right|$ the total frequency of all words in C_i . By considering the existence of documents in C_i , however, another approximate way for calculating $P(w_t^{C_i})$ can be achieved by applying the total probability theorem:

$$P(w_t^{C_i}) = P(w_t^{C_i} \cap C_i) = \sum_{j=1}^{n4} P(w_t^{D_j^{C_i}} | D_j^{C_i}) P(D_j^{C_i}). \quad (3.3)$$

where $P(w_t^{D_j^{C_i}} | D_j^{C_i})$ can be estimated by using a unigram language model the same way as in formula (3.2) by considering $D_j^{C_i}$ as "a bag of words". $P(D_j^{C_i})$ can be easily calculated as $1/|C_i|$, where $|C_i|$ denotes the number of documents contained in C_i .

The negative factor $P(w_t)$ in the denominator of formula (3.1) can be calculated in a similar way as in formula (3.3). As the entire population is C , the item $P(w_t)$ is equal to $P(w_t \cap C)$. Comparing C_i with $D_j^{C_i}$ and C with C_i , we have:

$$P(w_t) = P(w_t \cap C) = \sum_{i=1}^n P(w_t | C_i) P(C_i) . \quad (3.4)$$

Combining the estimations above we are able to calculate $P(C_i | w_t)$ in formula (3.1), i.e. the term occurrence of w_t for a certain category C_i .

3.3 Topicality-Based Term Weighting in a Pre-defined Category

Although term occurrence seems to be a good measure for term weighting, we felt that it doesn't take full advantage of the pre-defined category system. In a category system, documents are assigned to one or more categories by human experts. The topicality of documents to their host categories is guaranteed by human intelligence. This can, to some extent, help to assure the topicality of terms contained in these documents and facilitate its calculation.

A closer look at the estimation of $P(w_t^{C_i})$ in formula (3.2) and (3.3) reveals that its positive contribution to the term occurrence is actually a union effect of two aspects: the frequency of a term in a single document and the distribution of a term in the category. We believe that the second aspect, i.e. the term distribution in a category, plays a much more important role in deciding term topicality for this category than the first aspect. Intuitively, the more a non-stopword term is distributed in a category, the more it can represent the category and the more topical it is to the category. Instead, if another term occurs with rather high frequency only in a few documents in a certain category, it can possibly only represent these documents, not the whole category, although it is also likely to provide a high value of occurrence. Due to their different levels of representativeness to a given topic represented by a category, it is obvious that the first term should be assigned a higher topical weight than the second one in the category, which formulae (3.2) and (3.3) fail to do. In order to address this weakness, we conceptualize a new language model to give topical weight to term $w_t^{C_i}$.

$$TW(w_t^{C_i}) = \frac{|w_t^{C_i}| \times |D_{w_t}^{C_i}|}{|w^{C_i}| \times |D^{C_i}|} . \quad (3.5)$$

This formula consists of two factors which are multiplied together: a basis factor defined by $|w_t^{C_i}| / |w^{C_i}|$ and a modification factor defined by $|D_{w_t}^{C_i}| / |D^{C_i}|$, where $|D_{w_t}^{C_i}|$ is the number of documents in C_i containing the term w_t and $|D^{C_i}|$ denotes the total number of documents in C_i . $|w_t^{C_i}|$ and $|w^{C_i}|$ have the same meaning as defined in

formula (3.2). The idea is to adapt the calculation of $P(w_t^{C_i})$, which is an important factor for estimating term occurrence, to calculate term topicality. In order to keep the calculation simple, we choose the formula (3.2) as the basis factor and multiply it with the modification factor $|D_{w_t}^{C_i}| / |D^{C_i}|$, which indicates how widely $w_t^{C_i}$ is distributed in C_i . By introducing more impact through the modification factor and keeping its influence in the basis factor, the importance of term distribution in category is stressed. In this way, we believe that term topicality can be calculated more accurately. Because the new model expressed by formula (3.5) is a modified unigram model which is able to better describe the special distribution of a term in a pre-defined category, we call it a topical language model. As alternative to formula (3.2), the formula (3.3) can also be used as the basis factor in our topical language model. This will bring a slight improvement in calculating accuracy and a considerable increase of computing cost.

Let's now turn to $P(w_t)$ again, which is estimated by formula (3.4). As mentioned above, it provides a negative contribution to term occurrence. As term occurrence is somehow related to term topicality, we think it is necessary to do further investigation about it to explore which factors may negatively influence term topicality. A closer look at formula (3.4) implies that the more frequently a term in other categories appears, the lower its occurrence to the category C_i ; the more the term is distributed among different categories, the lower its occurrence to the category C_i . Compared to term occurrence, the calculation of negative factors for term topicality proves more complicated. The reason is the existence of multi-topicality, which means that a term could be topical in more than one categories. This implies that the occurrence of a term in one category will not necessarily reduce its topicality in another category. However, there is one fact that still holds: if a term is distributed too evenly among all categories, it has a too general meaning and is not specific to any category. Therefore, they should be given a low weight for their topicality to the category. We introduce the following formula to calculate how evenly a term is distributed among different categories.

$$\text{Degree of Distribution } dd = \sum_{i=1}^{n1} \left(\frac{|w_t^{C_i}| - MV}{S} \right)^2 = \frac{1}{S^2} \sum_{i=1}^{n1} (|w_t^{C_i}| - MV)^2 . \quad (3.6)$$

where $|w_t^{C_i}|$ is the frequency of term w_t in C_i , $i=1,2,\dots,n1$, $S=\sum_{i=1}^{n1} |w_t^{C_i}|$, $MV=S/n1$.

The degree of distribution measures the differences among the frequencies with which the term appears in different categories. The smaller the value, the more evenly the term is distributed. If the value is larger than a certain threshold (Th_1), it is considered that the term is not evenly distributed in the categories. In order to combine this factor with the weighting of term topicality, we make a rather rigid restriction that whenever the distribution grade of term is smaller than the threshold (Th_1), it will be assigned a weight of 0. That means the term will not be included in the final thesaurus.

Combining formula (3.5) and formula (3.6), we are now able to give our weighting scheme to calculate the topicality of a term to a specific topic represented by C_i .

In C_i , the topical weight of a term w_t can be calculated as following:

$$TW(w_t^{C_i}) = \begin{cases} \frac{|w_t^{C_i}| \times |D_{w_t}^{C_i}|}{|w_t^{C_i}| \times |D^{C_i}|} & \text{if } dd > Th_1 \\ 0 & \text{if } dd < Th_1 \end{cases} \quad (3.7)$$

4 Relationship Determination

The most intuitive way to calculate the mutual relationship between two random events is to calculate the mutual information between them. Pointwise mutual information is a special mutual information measure, which is used to calculate the association between two individual elements. We can then use it as an association measure between two terms in a corpus: $I(w_1, w_2) = \log_2(P(w_1, w_2) / (P(w_1) * P(w_2)))$. The larger the value of $I(w_1, w_2)$, the more w_1 and w_2 are related to each other. Let's take a closer look at this formula to see what it can reveal to us. For a certain w_1 , the contribution of w_2 to $I(w_1, w_2)$ depends on $P(w_1, w_2)/P(w_2)$. Intuitively, the more often w_1 and w_2 appear together, the bigger the value $P(w_1, w_2)$, the larger the value $I(w_1, w_2)$ will be. However, from the formula above, we can see a fact defying our intuition: low values of $P(w_1, w_2)$ can also lead to a high value of $I(w_1, w_2)$, if $P(w_2)$ is small enough. This fact reveals a new important relationship among words which are “related” to each other according to the point-wise mutual information measure. Let's imagine w_1 is a “big” word which appears very often in a category; w_2 is a rather “small” word which occurs seldom in the same category. However, if w_2 appears, it appears almost always together with w_1 in the same documents, while w_1 also appears in many other documents which don't contain w_2 . In this case, we could intuitively state that w_1 has a more “general meaning” than w_2 . In our work we call this new relationship the “Thematical Specialization” (TS) relationship. The other “traditional” relationship will be named as the “Mutually Relevant” (MR)-relationship. Since point-wise mutual information can only reflect the conjoint effect of TS- and MR-relationship, new measures should be found to calculate them separately.

We introduce two similarity grades, S_1 and S_2 . By applying a simple language model we have:

$$S_1 = P(w_2|w_1) = P(w_1, w_2)/P(w_1) = D(w_1, w_2)/D(w_1)$$

$$S_2 = P(w_1|w_2) = P(w_1, w_2)/P(w_2) = D(w_1, w_2)/D(w_2)$$

where $D(w_1, w_2)$ denotes the number of documents in the category which contains w_1 and w_2 and $D(w_j)$ denotes the number of documents which contain the term w_j .

A small value of the similarity S_1 with a much larger value of S_2 means that w_2 occurs relative frequently together with w_1 in the same documents while w_1 occurs only seldom with w_2 . w_2 is therefore probably a specialization term of w_1 . Two additional threshold values (Th_2 and Th_3) are introduced to determine how small S_1 and how big S_2 should be in order to identify the existence of the TS-relationship. With the help of these similarities and thresholds the term relationships in the thesaurus can be defined.

$$TS\text{-relationship: } \text{If } \begin{cases} S_1 \leq Th_2 \\ S_2 \geq Th_3 \end{cases} \text{ then } w_2 \text{ is a specialization term of } w_1.$$

This relationship shows that a topic, given by term w_1 , can be specialized or limited by the term w_2 . This is for example the case, if w_2 is a narrower term, a property, a part, or just a strongly associated term of w_1 .

The MR-relationship will be determined by the standard cosine measure. Another threshold value (Th_4) is introduced. Any term pair, whose cosine measure is larger than this threshold value, is regarded as mutually relevant, if there exists no TS-relationship between them. Let the similarity grade $S_0 = \cos(V(w_1), V(w_2))$. A large value of the similarity S_0 establishes the MR-relationship between w_1 and w_2 .

$$MR\text{-relationship: } \text{If } \begin{cases} S_1 > Th_2 \\ S_0 \geq Th_4 \end{cases} \text{ or } \begin{cases} S_2 < Th_3 \\ S_0 \geq Th_4 \end{cases} \text{ then } w_1 \text{ and } w_2 \text{ are mutually relevant.}$$

Other term pairs, which do not possess any of these two relationships, are regarded as mutually irrelevant.

$$Irrelevant: \text{ If } \begin{cases} S_1 > Th_2 \\ S_0 < Th_4 \end{cases} \text{ or } \begin{cases} S_2 < Th_3 \\ S_0 < Th_4 \end{cases} \text{ then } w_1 \text{ and } w_2 \text{ are mutually irrelevant.}$$

5 Clustering – The Layer-Seeds Method

Based on the term weighting scheme and the discussion of relationship determination in previous sections, we present in this section the complete algorithm of the Layer-Seeds Method for organizing terms in a hierarchical structure.

- a. At the very beginning only one layer exists, which contains all terms whose TW weights are greater than 0.
- b. A term in the current layer is selected as seed, on the condition that it is still not seed and its TW weight is the largest in this layer.
- c. The three similarities between the seed and all other non-seed terms in the same layer will be calculated and the relationship will be determined. Those terms, which have the TS-relationship with the seed, will be put into the next layer as specialization terms of the seed. This relationship will be named „direct thematical Specialization“. It still has to be checked whether these specialization terms also have the TS-relationship with the ancestors of the seed; if this is the case, the relationship with the ancestors will be named „indirect thematical Specialization“. Those terms that have a MR-relationship with the seed, will remain in the same layer and the relationship will be stored. Those terms that are irrelevant to the seed will not be handled and remain in the same layer.
- d. If non-seed terms in the current layer still can be found, step b will be repeated. If not and there still exist sub layers, the sequence number of the layer will be raised up by one and step b will be repeated; otherwise the whole process ends.

6 Experimental Results

The thresholds Th_1 to Th_4 , which play an important role in our approach, are determined empirically by experiments. Different threshold values are tried for totally 16 categories from Yahoo!, Dmoz and BOL. The results are generally satisfactory with $\text{Th}_1=0.1$, $0.3 < \text{Th}_2 < 0.5$, $0.7 < \text{Th}_3 < 0.9$, $0.05 < \text{Th}_4 < 0.2$, where Th_2 , Th_3 , Th_4 should be fine-tuned to adapt different categories.

The simplest way to evaluate an automatic constructed thesaurus is to compare it with a “gold standard”, i.e. one of the most famous manually created thesauri. Recent research [5, 11] shows that such comparison is not able to fairly judge the usefulness of automatic constructed thesaurus. As a matter of fact, one cannot even fairly judge between two manually created thesauri, because each individual creating a thesaurus is likely to create a different hierarchy based on personal biases. However, we think it is quite interesting to find out how our thesauri differ from those created manually. In our experiment we took the astronomy thesaurus¹ compiled for the International Astronomical Union as the referencing thesaurus. This manually created thesaurus contains about 3000 concepts and 3 kinds of relations: bt/nt (broader term/narrower term), rt (related term) and u/uf (use/use for). Our thesaurus is built automatically upon the astronomy category of yahoo!². We crawled all websites collected in this category and its subcategories. In each website only the summarization information (title, keywords and description) in the first page directly linked by yahoo! is retrieved. In the final thesaurus we have around 5300 concepts with its $TW>2$ and two kinds of relationships as described in previous sections: TS- and MR-relationship. For the comparison, we first chose a total of 200 concepts randomly from the manually created thesaurus, where 97 of them are one-word concepts, 85 of them two-words concepts and 18 of them three-words concepts. Comparison shows that approximately 85% of the one-word concepts are also contained in our automatic built thesaurus; the results for the two-words and three-words concepts are 35% and 22% respectively. For comparing relation between concepts we chose randomly 100 bt/nt relations and 100 rt-u/uf relations from the manually created thesaurus. They are then compared with the TS- and MS-relations in our thesaurus respectively. The corresponding results are 11% and 16%.

As mentioned above, the large difference between the two thesauri doesn't indicate the usefulness of both thesauri. The judging criteria for usefulness of a thesaurus vary for different users in different application situations. We take the concept “Radiation” as an example, which appears in both thesauri. A part of its narrower terms in the manually created thesaurus is listed below: BACKGROUND RADIATION, BLACK BODY RADIATION, COSMIC RAYS, GAMMA RAYS, GRAVITATIONAL RADIATION, HIGH ENERGY RADIATION, INFRARED RADIATION, MICRO-WAVE RADIATION, POLARIZED RADIATION, RADIO RADIATION, SYNCHROTRON RADIATION, THERMAL RADIATION, X RAYS. Several important TS-terms to the concept “Radiation” in our thesaurus are also listed as follows: pulsar radiation, heat radiation, radiation environment, radecs, background radiation, Space Radiation Laboratory, circuit, microelectronic devices, NSREC, infrared radiation.

¹ English version 1.1. Compiled by Robyn M. Shobbrook and Robert R. Shobbrook.
ftp://ftp.aoe.gov.au/pub/lib_thesaurus/trex.zip

² <http://dir.yahoo.com/Science/Astronomy/>

Apparently, although many of our TS-terms are specific to the domain astronomical radiation, they are not strict narrower terms according to the traditional scheme of astronomical classification. However, they do provide other interesting points of view for this domain. A user, for example, who is interested in research activities about radiation in the field astronomy, will find our thesaurus useful since it contains the names of different relevant conferences and workshops, like “NSREC” (Nuclear and Space Radiation Effects Conference) and “radecs” (Radiation and its Effects on Components and Systems).

An alternative and more accurate way for evaluating an automatically constructed thesaurus is to involve it in a special task and see to how well it is able to help performing the task. In an e-commerce project funded by EU (COGITO [10]), we were able to build a thesaurus automatically with the “Layer-Seeds Method” based on the information about books provided by an online bookstore --- BOL Germany. We applied the thesaurus for automatic query expansion to investigate how well the thesaurus helps to enhance the system performance and user satisfaction. The experiment used the BOL’s German book category system by considering the topic “Computer & Internet”. It contains about 500 book titles and related descriptions, further subdivided in 9 categories. The final thesaurus contains about 2000 concepts in German language. As an example of the thesaurus we present a part of the hierarchical structure for the seed “Java” in category “Programming Language” in table 1; in table 2, the top twenty most relevant terms to the seed “Windows” in category “operating system” are listed in descending order according to their S_0 values. All German terms in the tables are translated into English.

Table 1. A part of hierarchical structure of term java

1:Java		
_2:Netscape	_2:RMI	_2:JDK
_3:Browser	_3:programming	_3:Security
_3:VBScript	_3:Security	_3:Class library
_3:Stylesheet	_3:Object orientation	_3:Distribution
_3:JavaScript	_3:JavaBeans	_3:Java2D
_3:Information	_3:Accessibility	_3:IDL
_3:User	_3:Distribution	_3:Application
_2:Applets	_3>Type transformation	example
_3:Threads	_3:Toolkit	_2:JDOM
_3:Security	_3:Thread programming	_2:API
_3:Accessibility	_3:Superclass	_2:Basic concept
_3:Toolkit	_3:Subclass	_2:Corba
_3:Thread programming	_3:Software solution	_2:Propositional logic
_3:Superclass	_3:Package	_2:Web programming
_3:Subclass	_3:OMG	_2:Unicode
_3:Method call	_3:Method call	_2:Threading
_3:Convention	_3:Class declaration	_2:Platform
_3:Class declaration	_3:IDL	_2>Data structure
_3:JFC	_3:Area of validity	_2:Expression
	_2:Network	_2:Mail

Table 2. The most relevant terms to windows

Windows
Microsoft
learn
network
setup
configuration
operating system
multimedia
programming
Kit
work
install
handbook
user
home user
application
linux
beginner
basis
interface
registry

The main task of the online bookstore system is to help user in quickly finding the needed books. We applied our thesaurus to expand unsuitable user queries to retrieve better search results. By comparing the retrieval performance and user satisfaction before and after using the thesaurus, the “usefulness” of the thesaurus can be judged. The thesaurus built by the Layer-Seeds Method can be used either for automatic query

expansion by applying the MS-relationship or interactive query expansion by applying the TS-relationship. In an e-commerce system, most of the users are not search experts and not always willing to make long interaction with the system for choosing the most suitable keywords. They just wish to find the satisfactory products as quickly as possible without any extra burden. Therefore, in our online bookstore application, only the MS-relationship was used for automatic query expansion. We plan to test the TS-relationship in future in other more suitable applications. In the experiment, we applied HySpirit [2] as the underlying search engine. The evaluation procedure consists of three steps. A baseline was first built by evaluating the result sets of 60 original single-term queries, which were chosen randomly from the thesaurus. In a second step, the original queries were expanded with their most similar terms according to the MR-relationship in the thesaurus. The result sets of these expanded queries were evaluated the same way as for the baseline. Finally we further expanded the original queries with their second most similar terms and evaluated the result sets analogously to the previous steps. The evaluation of the three result sets was carried out by estimating precision and recall for each query. Since the retrieval process is embedded in an interactive e-commerce dialogue, it is reasonable to restrict the amount of hits shown to the user. Therefore, we used user-oriented performance measures. Precision and recall were computed taking the top n ($n < 11$) elements from the result lists. Obviously, this leads to high precision estimates --- as the first results can be expected to be relevant --- but decreases the estimated recall values (relevant items which are not among the top n elements are neglected). Experimental results show that both the query expansion steps on the basis of the thesaurus did not produce a significant deviation from the baseline in terms of precision but achieved an improvement in terms of recall (29,02% for the first and 5% for the second expansion step), which meets the requirements set up for the thesaurus-based expansion method in our e-commerce application context. Figure 1 shows a P-R graph containing three trend lines (polynomial regression with 4 as the highest exponent) for each result set --- baseline, first expansion and second expansion. The white curve, which represents the first expansion result, lies clearly above the black curve, which represents the baseline. The expansion model presented above has been also evaluated as a whole within the project COGITO. Report shows that, with help of our thesaurus, a total of 61% increasing of user satisfaction was achieved. (cf. [10] for more details).

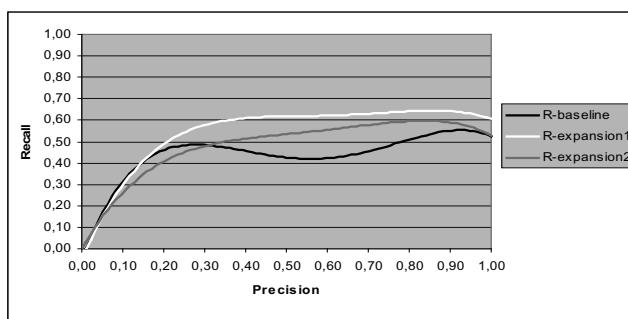


Fig. 1. The precision-recall graph

7 Conclusion and Future Work

In this work we proposed an approach to construct domain thesauri automatically based on pre-defined category systems. A topical language model is conceptualized to calculate term topicality for a certain category. Relationships among thesaurus elements are determined by analyzing the pointwise mutual information measure. We presented a clustering method --- the “Layer-Seeds method” to organize terms in a certain category to a hierarchical topical structure. The thesaurus contains satisfactory structures, although it differs to some extent from a manually created thesaurus. The usage of such a thesaurus for automatic query expansion enhanced performance of information retrieval and user satisfaction.

In the future we plan to further investigate the characteristics of different category systems and how they could be effectively combined. We further intend to apply the TS-relationship in the thesaurus for interactive query expansion. Application of the thesaurus for document classification as shown in [3] is also of our future interest.

References

1. Crouch, C.J., Yang, B., Experiments in automatic statistical thesaurus construction, SIGIR'92, 15th Int. ACM/SIGIR Conf. on R&D in Information Retrieval, Copenhagen, Denmark, 77-87, June 1992.
2. Fuhr, N. and Roelleke, T. HySpirit --- A probabilistic inference engine for hypermedia retrieval in large databases. *International Conference on Extending Database Technology*, Valencia, Spain, 1998.
3. A. Gelbukh, G. Sidorov, A. Guzman-Arenas. Use of a weighted topic hierarchy for document classification. In: Text, Speech and Dialogue. Lecture Notes in Artificial Intelligence, N 1692, Springer-Verlag, 1999.
4. Jing, Y. F. and Croft, W. B.: An Association Thesaurus for Information Retrieval, In RIAO 94 Conference Proceedings, p. 146-160, New York, Oct. 1994.
5. D. Lawrie. Language Models for Hierarchical Summarization. Dissertation. University of Massachusetts, Amherst, 2003.
6. Qiu, Y., Frei, H.P.: Concept based query expansion. In Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval, ACM Press, 160-170, 1993.
7. G. Salton,, Automatic Information Organization and Retrieval, McGraw-Hill Book Company, 1968.
8. M. Sanderson & B. Croft. Deriving concept hierarchies from text. In the Proceedings of the 22nd ACM SIGIR Conference, Pages 206-213, 1999.
9. Sparck-Jones, K.: Automatic Keyword Classification for Information Retrieval. Butterworth, London, 1971.
10. Thiel, U.; L'Abbate, M.; Paradiso, A.; Stein, A.; Semeraro, G.; Abbattista, F.; Lops, P.: The COGITO Project. In: e-Business applications: results of applied research on e-Commerce, Supply Chain Management and Extended Enterprises. Section 2: eCommerce, Springer, 2002.
11. Kilgariff, A. Thesauruses for Natural Language Processing. Technical Report Series: ITRI-03-15, ITRI, Univ. of Brighton.

Populating a Database from Parallel Texts Using Ontology-Based Information Extraction

M.M. Wood¹, S.J. Lydon², V. Tablan³, D. Maynard³, and H. Cunningham³

¹ Dept of Computer Science, University of Manchester, Manchester, UK
mary@cs.man.ac.uk

² Earth Science Education Unit, Keele University, Staffordshire, UK
s.j.lydon@educ.keele.ac.uk

³ Dept of Computer Science, University of Sheffield, Sheffield, UK
{valyt,diana,hamish}@dcs.shef.ac.uk

Abstract. Legacy data in many mature descriptive sciences is distributed across multiple text descriptions. The challenge is both to extract this data, and to correlate it once extracted. The MultiFlora system does this using an established Information Extraction system tuned to the domain of botany and integrated with a formal ontology to structure and store the data. A range of output formats are supported through the W3C RDFS standard, making it simple to populate a database as desired.

1 Overview

Legacy data in botany, as in other mature descriptive sciences, is distributed across multiple text descriptions: the record of careful observations of plant groups (taxa), repeated across time and space. These essential information resources for biodiversity research are cumbersome to access - requiring patient hours in large libraries - and fixed in a pattern designed to support identification of specimens, and unsuited to other purposes.

The MultiFlora system has adapted standard techniques for Information Extraction to botanical text, and integrated an ontology for plant anatomy, which is used to store the extracted information and correlate it across multiple source texts. The result is a structured, flexible data resource, which can be exported to a range of database formats. The domain-tuning is carefully localised to facilitate adapting the system to other areas.

1.1 Parallel Texts

“Parallel texts” are the focus of lively current research interest, and the phrase is used to describe various lines of research, including cross-document co-reference [1], summarisation of multiple documents [7,9]), and the processing of translation equivalent texts (as in the CLEF Medical Informatics project¹). We are doing none of these. We are analysing heterogeneous, monolingual (English) texts

¹ <http://www.clinical-escience.org/>

which aim to convey the same information. Merging the information extracted from several such texts compensates both for gaps in the texts themselves and for failures in processing. The results presented below demonstrate the value of this approach.

1.2 Ontology-Based Information Extraction

Established techniques for Information Extraction have been developed for relatively small and simple test domains, and hold the information extracted in a flat, or slightly structured, “template” format. Extending these techniques to a complex, real-world domain such as botany requires a major step up in the quantity and complexity of information to be extracted and represented. When our project set out to acquire, in flexible electronic format, some of the wealth of legacy data locked in botanical descriptive texts (Floras), we rapidly reached the limits of the template format, such as that used in MUC [2], finding it cumbersome and insufficiently expressive for our domain. We have therefore integrated the established GATE natural language processing system [3] with a range of tools for building and using ontologies, including Protege² for ontology development, and a Sesame³ knowledge repository to store ontological data.

2 Data Distribution in Parallel Texts

Our initial data set was made up of the descriptions of five species of *Ranunculus* (buttercup: *R. bulbosus*, *R. repens*, *R. hederaceus*, *R. acris*, *R. sceleratus*) given in six Floras, chosen to be both representative of the many available, and independent of each other. As a test set we used descriptions of five species of the Lamiaceae (mints and related herbs: *Mentha spicata*, *Nepeta cataria*, *Origanum vulgare*, *Salvia pratensis*, *Thymus serpyllum*). The text-type is syntactically identical to the training set, while the plant morphology and associated vocabulary are sufficiently different to test the scalability of our approach.

Most botanical descriptions are not true “natural” language, but highly stylised lists of noun phrases, usually lacking verbs and articles altogether. This leads to an information-dense text type (a result of the space limitations historically imposed on authors of Floras). Here is a short, but otherwise typical example:

1. *R. acris* L. - Meadow Buttercup. Erect perennial to 1m; basal leaves deeply palmately lobed, pubescent; flowers 15-25mm across; sepals not reflexed; achenes 2-3.5mm, glabrous, smooth, with short hooked beak; 2n=14. [8]

Although problematic for a general-purpose parser, this text-type is highly amenable to analysis with specifically tuned resources.

² <http://protege.stanford.edu/>

³ <http://www.openrdf.org/>

Based on hand analysis of the texts, we built a three-dimensional data matrix for each species, where columns represent Floras, rows represent characters, and the third dimension is the taxonomic hierarchy, allowing inheritance of information from higher taxa into species descriptions where appropriate.

Table 1. Fragment of the information about the leaves of *R. acris* given by four authors

Heading	1	2	3	4
Basal leaves	Basal lvs	Basal leaf	rosette-leaves	basal leaves
Petiole				
presence	-stalked		-petioled	
length	long		long	
Lamina		blades		
width		1.6-5.2 cm		
length		x 2.7-5.6		
surface				pubescent
divisions	-lobed	-parted	-parted	lobed

The matrix is sparsely populated: it is common for only one of the six Floras to give any value for a character. Thus, for example, the six descriptions of *R. acris* have, on average, 44 slots each in their individual templates (range 17-68). The combined template for *R. acris* from all sources has 134 slots.

The results of our hand analysis of data distribution in the initial texts are presented in full in [4]. In summary, more than half of the data items in the final set are found in only one of our six sources, while disagreement accounted for only 1%.

agreement	9%
overlap	36%
disagreement	1%
single source	55%

(The example above shows agreement in one out of nine slots - the use of “long” by authors 1 and 3 - and overlap in three - the synonym sets “basal leaves” / “rosette-leaves”, “-stalked” / “-petioled”, and “lobed” / “parted”. Four are single-source, and one is empty.)

The results of correlating output from the initial prototype MultiFlora IE system are presented in full in Wood et al (in press). We show there that recall summed over six botanical descriptions of several plant species is more than triple the average for each text individually. This arises from two independent factors:

- (i) The source texts show the “sparse data matrix” property just referred to;
- (ii) Failures in extraction from any one text are significantly often corrected

by accurate results from other texts. 50% of all false negatives are compensated by results from other sources; in 24% of template slots, a false negative in processing one text is compensated by accurate processing of one or more others.

3 Ontology-Based Information Extraction

3.1 System Architecture

The Information Extraction sub-system of our application is capable of finding *heads* and *features* in the input texts using a combination of techniques ranging from keyword search to internal and contextual pattern searching and shallow linguistic analysis based on information such as the part of speech.

The system builds on the Natural Language Processing capabilities provided by GATE, a General Architecture for Text Engineering, developed by the NLP group at the University of Sheffield [3]; it is based on ANNIE, an information extraction system integrated into GATE. Eight processing resources are used. The first five - the Document Resetter, Tokeniser, Abbreviations, Sentence Splitter, and POS tagger - are standard components apart from some additional vocabulary in the POS Tagger. The last three - the Ontological Gazetteer, JAPE Transducer (holding the grammar rules), and Results Extractor - are more or less specific to the domain.

3.2 Ontology-Based IE

Because of the complexity of the information to be extracted, the template model traditionally used in IE [2] was replaced with an ontology-based method. This provides richer information structures and greater flexibility than a fixed template. The system starts with a base upper ontology (developed and debugged in Protege), containing object classes related to the botanical domain and template properties that can hold between them. The output of the IE system is a copy of the base ontology populated with object instances, i.e. the heads and the features found by the textual analysis, and with property instances, i.e. relations between the object instances found in the text.

The discourse modelling - which has the role of linking the features to the right head by finding relations - is performed simultaneously with the ontology population, as it was expected that the two operations can mutually inform each other. Because the texts use a very restricted syntax, the discourse modelling cannot be based on parsing the input, and had to rely on heuristics instead. We found that the language used in the botanical texts, while characterised by a very high informational content, has a very simple and flat discourse structure. Each text consists of a sequence of segments which are, in most cases, formed by a head followed by a list of related features: e.g. “Achenes in a subglobose head, flat, margined, broadly obliquely obovate, 2-3 mm. long; beak 0.4-1 mm. long.”. There are some exceptions where some of the features can occur in the text before the head, and other situations where the head is omitted (e.g. when the

first segment of the text simply lists a set of features relating to the plant being described: e.g. "Perennial 10-100 cm, more or less glabrous, appressed-pubescent or with stiff patent or deflexed hairs on stem."

The process of analysing each text starts with a copy of the base ontology, which was created to model the botanical domain, and contains about 70 classes representing heads and features with around 20 properties between them. The ontology was developed using Protege, incorporating a combination of information found by hand analysis of the test set and in a comprehensive botanical lexicon [5]. Heads are things like plants, plant parts and organs, while features represent characteristics such as shape, colour or size. Properties are used to link heads to features and encode the fact that certain features are applicable to particular heads. Fragments of the base ontology are shown in Figures 1, 2 and 3.

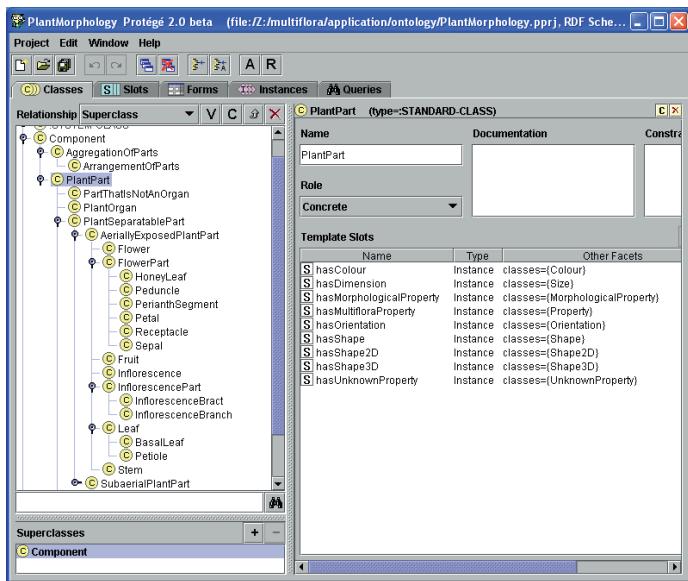


Fig. 1. Ontology Heads

Before starting to populate the base ontology with instances, discourse segments need to be identified. For this we are using a very simple algorithm which breaks the text into segments whenever it encounters sentence ending punctuation (such as full stops) or semicolons. The resulting segments are expected to contain a single head and a set of compatible features for it. Although rudimentary, this procedure is successful in most of the cases because of the regular nature of the texts. The cases where it fails, for instance when the author uses commas to separate segments, tend to be corrected by the downstream process

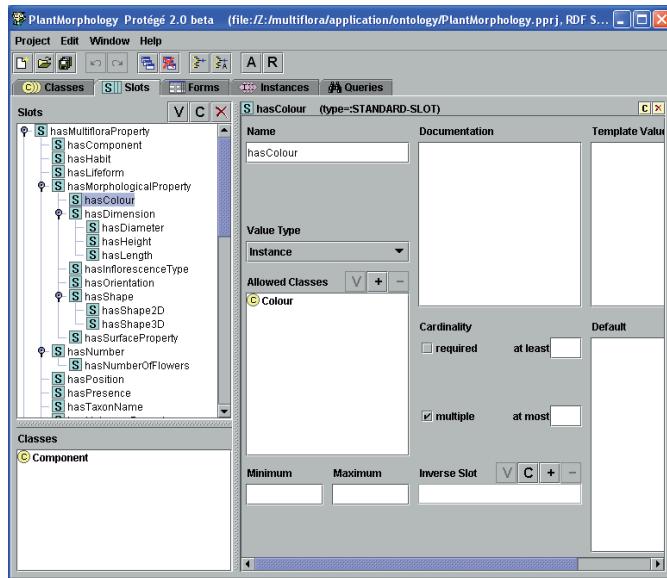


Fig. 2. Ontology Properties

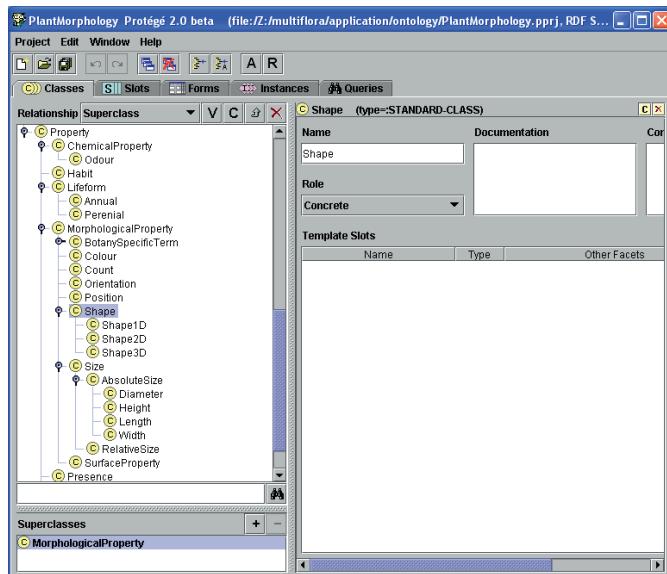


Fig. 3. Ontology Features

which automatically recognises a new segment whenever a new head is mentioned regardless of whether a separator was identified or not. The combined use of separators and expectations regarding the internal structure of segments leads to a good segmentation of the text.

The next step is to link the heads and the features identified by the IE sub-system with the base ontology and to generate instances for populating the ontology. This is done using an *ontological gazetteer* containing domain-specific terms. An ontological gazetteer is a list lookup component that links terms with classes in an ontology. The one used in our system lists more than 2000 entries in 43 separate lists (each list being linked to a class in the base ontology).

Head categories include specific plant parts:

Flower(*flower, floret, fl*)

Leaf(*leaf, leaves, fronds*)

Petal(*petal, honey-leaf, vexillum*)

and also collective categories such as

PlantSeparatablePart(*appendage, tuber*)

PlantUnseparablePart (*beak, lobe, segment*)

SpecificRegionOfWhole (*apex, border, head*)

Features include plant characteristics such as:

Habit (*bush, shrub, succulent*)

MorphologicalProperty (*dense, contiguous, separate*)

SurfaceProperty (*pilose, pitted, rugose*).

Properties are often more general:

2DShape (*arching, linear, toothed*)

3DShape (*branching, thickened, tube*)

Colour (*glossy, golden, greenish*)

Count (*numerous, several*).

This initial step linking heads to features introduces a relatively large amount of ambiguity, because some terms appear in more than one list, or because some heads or features in the text contain more than one term listed in the gazetteer. This ambiguity is resolved by the final processing resource in the application - the results extractor. This last PR also deals with validating the results, disambiguating and adding the instances into the ontology and generating the appropriate properties between the newly created instances.

The results extractor uses a Sesame semantic repository to store the evolving ontology and to perform basic queries against it for validating the results. Disambiguation between potential classes for instances found in the text is done using a heuristic that prefers the more specific against the more generic ones. Once the classes for the head and associated features have been determined, the properties that are compatible with each head-feature pair are enumerated and, again, the more specific ones are preferred. The resulting <head, property, feature> triples are added into the ontology.

We use the WWW Consortium standard RDFS (Resource Description Framework Schema⁴) format, designed to “provide interoperability between ap-

⁴ <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>

plications that exchange machine-understandable information on the Web”, for the input/output and storage of knowledge, which makes it possible to combine all these tools (GATE, Sesame, Protege and a variety of other knowledge editors). Results are currently exported into XML markup and Excel spreadsheets; the use of RDFS ensures that a range of alternative export formats can easily be supported.

4 Results

A typical sample of input text for Ranunculus is:

Perennial herb with overwintering lf-rosettes from the short oblique to erect premorse stock up to 5 cm, rarely longer and more rhizome-like; roots white, rather fleshy, little branched.

The output data for this text is shown in Table 2. A typical fragment of ontology is shown in Figure 4.

Table 2. Output data for Ranunculus

Head Class	Head	Property	Feature Class	Feature
Plant	herb	hasLifeform	Lifeform	Perennial
Leaf	lf-rosettes	hasLifeform	Lifeform	overwintering
PlantSeparatablePart	stock	hasRelativeProperty	RelativeProperty	short
PlantSeparatablePart	stock	hasOrientation	Orientation	oblique to erect
PlantSeparatablePart	stock	hasLength	Length	up to 5 cm
PlantSeparatablePart	stock	hasRelativeProperty	RelativeProperty	rhizome-like
Root	roots	hasColour	Colour	white
Root	roots	hasShape3D	Shape3D	rather fleshy
Root	roots	hasShape3D	Shape3D	little branched

Although some information has been lost in processing (most importantly “rarely longer and more...” modifying “rhizome-like”), most has successfully been extracted from the text into a principled, structured format which can be exported to a wide range of databases and other applications. Further refinement of our text processing functionality will improve the results of extraction; also, the redundancy in the parallel text sources offers a worthwhile chance that the gap will be filled from another text.

As mentioned above, empirical testing of the initial prototype MultiFlora IE system [10] showed that recall summed over six botanical descriptions of several plant species is more than triple the average for each text individually, both because of the “sparse data matrix” property of the texts, and because failures in extraction are indeed significantly often corrected by accurate results from other texts. 50% of all false negatives are compensated by results from

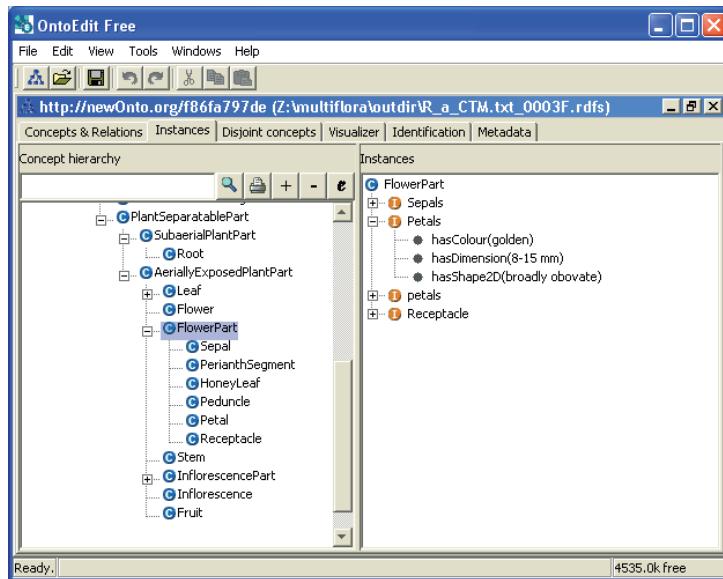


Fig. 4. Results for *Ranunculus* in OntoEdit

other sources; in 24% of template slots, a false negative in processing one text is compensated by accurate processing of one or more others.

Results on the test set of Lamiaceae descriptions found a small number of “UnknownPlantParts” which are not found in *Ranunculus*: floral structure differs significantly in the mints from the simple buttercup, so new terms and features (e.g. “calyx”) must be introduced. (There are around 110 slots for inflorescence in *Ranunculaceae*, around 190 in Lamiaceae). Thus extending the coverage of the system to new plant taxa requires some addition to the system, but our ontology-based design makes it the work of, literally, a few seconds to make the required addition in a principled way, by adding the necessary terms to the appropriate ontological gazetteer lists.

Overall, the MultiFlora project has provided successful proof of concept that ontology-based information extraction can address multiple parallel text sources for important legacy data in a specialised technical domain, and return that data in a flexible, structured format available for export to populate a range of databases.

5 Prospects

Botany is typical of many important technical domains in having valuable legacy data unhelpfully distributed over multiple natural language text sources. In

Bioinformatics, for example, new research results are appearing rapidly in journals and in the comment fields of various large databases (e.g., in 2000, 130 publications per month on *E. coli* metabolism). Keeping up with these developments currently requires massive human effort. SWISS-PROT is probably the largest, most accurate, and most important of the databases in this field:

SWISS-PROT... depends on a vast human network to generate its annotation by systematically running analysis software, reading the literature, contacting experts in the field, and so on. SWISS-PROT annotators probably represent the largest body of individuals in the bioinformatics community dedicated to the maintenance of a single public resource... Clearly, there is a demand for information that helps the consumer turn data into understanding ... the challenge is to make this kind of human effort sustainable [6].

This domain shares with botany all the characteristics which support our multiple-source, ontology-based techniques. The volumes of text to be processed in this domain are larger than in botany, but they already electronic, so our only real bottleneck, hand correction of OCR output, is not an issue.

Re-tuning the system to a new domain is largely a matter of replacing those ontological gazetteer lists which are specific to botany with whatever is desired. (Many of the lists are in fact general-purpose and re-useable, such as those for colour and measurement.) Depending on the text-type of the new domain, it may also be necessary to modify the JAPE Transducer which holds the grammar rules. Thus our prototyped model of ontology-based information extraction from parallel texts as a means of providing data to populate a database in a large technical domain is both feasible on a significant scale, and of significantly wide value.

References

1. Bagga, A., Biermann, A. W.: A Methodology for Cross-Document Coreference. Proceedings of the Fifth Joint Conference on Information Sciences (2000) 207-210.
2. Chinchor, N.: MUC-4 Evaluation Metrics. Proceedings of the Fourth Message Understanding Conference (1992) 22-29.
3. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics. (ACL-02), Philadelphia, USA (2002).
4. Lydon, S.J., Wood, M. M., Huxley, R., Sutton, D.: Data Patterns in Multiple Botanical Descriptions: implications for automatic processing of legacy data. Systematics and Biodiversity **1(2)** (2003) 151-157.
5. Lawrence, G.M.H. Taxonomy of Vascular Plants. Macmillan, New York (1951).
6. Miller, C.J., Attwood, T.K.: Bioinformatics goes back to the future. Nature Reviews Molecular Cell Biology **4** (2003) 157-162.
7. Radev, D.R., McKeown, K.R.: Generating Natural Language Summaries from Multiple On-Line Sources. Computational Linguistics **24 (3)** (1998).

8. Stace, C. *New Flora of the British Isles*. Cambridge University Press (1997).
9. Stein, G.C., A. Bagga, Bowden Wise, G.: Multi-Document Summarization: Methodologies and Evaluations. *Proceedings of the 7th Conference on Automatic Natural Language Processing* (2000) 337-346.
10. Wood, M.M., Lydon, S. J., Tablan, V., Maynard, D., Cunningham, H.: Using parallel texts to improve recall in IE. *Recent Advances in Natural Language Processing: Selected Papers from RANLP 2003*, John Benjamins, Amsterdam/ Philadelphia (in press).

A Generic Coordination Model for Pervasive Computing Based on Semantic Web Languages

Amine Tafat, Michele Courant, and Beat Hirsbrunner

Pervasive and Artificial Intelligence Group,
Computer Science Department,
Fribourg University Postfach 10 52 80,
1700 Fribourg, Switzerland

{Amine.Tafat-Bouzid, Michele.Courant, Beat.Hirsbrunner}@unifr.ch
<http://diuf.unifr.ch/pai/>

Abstract. Human interaction occurs always in a specific context and in a particular environment, and a common knowledge base about them is essential for understanding each other. By immersing computational system into physical world, pervasive computing bring us from traditional desktop computing interaction, to a new paradigm of interaction closer to Humans one's in term of context and environment dependency, and knowledge sharing.

To tackle this problem, we present in this paper, XCM, a generic coordination model for Pervasive Computing. XCM is organized around a few abstract concepts (entity, environment, social law and port), and is expressed as an ontology by using semantic web languages. While the abstract concepts of XCM deal with environmental representation and context-dependency, the use of the semantic web language OWL allows us to achieve knowledge sharing and context reasoning within Pervasive Computing coordination.

Keywords: Pervasive Computing, Ontology, Semantic Web, Coordination Model, context-awareness, knowledge sharing, OWL.

1 Introduction

Computing is moving toward pervasive environments in which devices, software agents, and services are all expected to seamlessly integrate and cooperate in support for human objectives - anticipating needs, negotiating for service, acting on our behalf, and delivering services in anywhere any-time fashion [1].

Pervasive environments can be considered as physical environments saturated with computing and communication, yet gracefully integrated with human users. In the future our daily environment will contain a network of more or less specialized computational devices that interact among themselves and with us.[4]

However, the tendency of merging physical world and computational systems, requires an appropriate software infrastructure and development tools. In particular, from coordination and integration perspective, the use of computers in non-desktop environments leads to go beyond traditional desktop interaction paradigm.

When observing interaction between humans it appears that communication is influenced by context and environment. The situation in which the communication takes place provides a common ground. This common ground generates implicit conventions, which influence and to some extent set the rules for interaction and also provide a key to decode meaning of words and gestures [2] Moreover, a common knowledge base is essential for understanding each other.

Thus, in order to successfully embed pervasive computing capabilities in our everyday environments, we need to define a coordination paradigm that take into account the two humans interactions characteristic cited earlier: Context-awareness, and knowledge sharing.

In this paper we propose a generic coordination model XCM based on:

- Reflective approach, that models and represents physical world. Embedding physical world representation into applications allows to enhance pervasive computing components with contextual information, achieving context modelling and making applications context-aware.
- Semantic Web [12], to express the modeled contextual informations as an ontology through the Ontology Web Language[20]. The ontological representation of context allows us to achieve contextual knowledge sharing and context reasoning between applications partners.

The present paper is organized as follows. In the next section we overview the concept of coordination model, discuss requirements for pervasive computing coordination model, and the gain obtained from using semantic web. Section 3 describes the concepts of abstract model XCM, and its ontological representation with Ontology Web Language OWL. In Section 4 we discuss related works, and in the last section we conclude this document.

2 Coordination and Coordination Model

Coordination can be defined as the process of managing dependencies between activities [5], or, in the field of Programming Languages, as the process of building programs by gluing together active pieces [6]. To formalize and better describe these interdependencies it is necessary to separate the two essential parts of a distributed application namely, computation and coordination. This sharp distinction is also the key idea of the paper of Gelernter and Carriero [6] where the authors propose a strict separation of these two concepts. The

main idea is to identify the computation and coordination parts of a distributed application. Because these two parts usually interfere with each other, the semantics of distributed applications is difficult to understand.

A general coordination model is usually based on three components [7] :

- Coordination entities, as the processes or agents which are subject of coordination;
- Coordination medium, the actual space where coordination takes place;
- Coordination laws, to specify interdependencies between the active entities;

As pervasive computing is nothing than computing in the context [2], it is essential to the intended coordination model to focus onto the context-sensitivity of the manipulated entities. The model must represent the context, and enable sharing contextual knowledge between interacting entities.

In order to achieve this purpose, it requires contextual information to be represented in ways that are adequate for processing and reasoning. Semantic Web [12] provides a suitable technology to define ontologies for modeling context, providing common vocabularies for knowledge sharing. Our proposal is to use OWL language [20] to define a coordination model for pervasive computing called XCM.

3 XCM, Generic Coordination Model

XCM is a coordination model designed to support the specificity of a pervasive application. Its essential characteristics are:

- Genericity, which is obtained through a high level of abstraction based on the notion of entity and agent;
- Capacity to handle the dynamics of ubiquitous execution environments - either they are physical or virtual-, and the context-sensitivity of applications, thanks to the explicit notion of environment;
- Homogeneous management of the contextual dynamics of components by the unique formalism of social law attached to the notion of environment, and a mechanism of port allowing entities to interact both very flexibly and powerfully.

As a coordination model, XCM comes within P. Ciancarini's approach [7], and the vision of coordination proposed by T. Malone [5], while prolonging an experience of coordination platform development we had previously carried on [9]. Within this approach, however it adds on a theoretical component inspired by autopoiesis i.e. the modeling of living systems elaborated by F. Varela and H. Maturana [10]. The interest of autopoiesis heritage is double. First, it allows profiting from the specificity of the physical space for modeling mechanisms like the dynamic management - namely the construction and the maintenance-

of organism frontiers. Second, it introduces a fundamental distinction between organisation (domain of control expression) and structure (domain of entity existence).

3.1 Entities

Everything is an entity in XCM. An entity e_i is defined by its structure, which is expressed as a recursive composition of entities $e_{i1} \dots e_{in}$ -called components of e_i - and by its organisation.

When modeling the entity concept as an ontology, we define a class called **Entity**. This abstract class defines a set of properties that are common to all entities, which consists of **HasStructure** and **HasOrganisation**. \\ \verbEntity+ classes have associated containment relationships. The relationships are defined by two related object properties called **HasComponents** and **IsComponentOf**. **HasComponent** property describes that the subject of this property is composed by the object of this property, and describes the subject of this property is a component of the object of this property. In the context of the OWL language, these two properties are defined as an inverse property of each other, as shown in the partial XCM ontology code bellow:

```

<owl:Class rdf:id="Entity">
  <owl:UnionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#AtomicEntity"/>
    <owl:Class rdf:about="#CompoundEntity"/>
  </owl:UnionOf>
</owl:Class>

<owl:ObjectProperty rdf:id="HasComponents">
  <rdfs:domain rdf:resource="#Entity"/>
  <rdfs:range rdf:resource="#Entity"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:id="IsComponentOf">
  <owl:inverseOf rdf:resource="#HasComponents">
    <rdfs:domain rdf:resource="#Entity"/>
    <rdfs:range rdf:resource="#Entity"/>
  </owl:inverseOf>
</owl:ObjectProperty>
```

Atomic entity. An entity, whose structure can not be decomposed, is called atomic; it denotes a pre-constructed element of the system. Whereas, the highest-level entity recursively containing all the other entities of the system, is called the universe of the system.

As atomic entity do not contain other entities, we introduce an abstract class called **AtomicEntity** which inherits all properties from its superclass

Entity while adding restrictions on the range of HasComponents property. In **AtomicEntity** class, the cardinality of the property HasComponents is 0 indicating all instances of this class do not contain any other entity. On the other hand, **CompoundEntity** is introduced to represent a set of entities that contains at least one Entity class member. **CompoundEntity** inherits all properties from **Entity** class with restrictions on minimal cardinality of HasComponents property. Partial XCM ontology code is presented here:

```
<owl:Class rdf:id="AtomicEntity">
  <rdfs:subClassOf rdf:resource="#Entity"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:OnProperty rdf:resource="#HasComponents"/>
      <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">0
      </owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassof>
</owl:Class>

\vspace*{-3pt}<owl:Class rdf:id="CompoundEntity">
  <rdfs:subClassOf rdf:resource="#Entity"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:OnProperty rdf:resource="#HasComponents"/>
      <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
      </owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassof>
</owl:Class>
<Entity rdf:id="Universe" />
\vspace*{-3pt}
```

The organisation of an entity e_i specifies the rules governing the assembling of components in the structure and their dynamics. Then it characterizes the domain of the interactions that are applied to e_i . It is expressed as a set of rules called by extension the social laws of e_i

3.2 Environment and Social Laws

At a given moment of the existence of the system, every entity e_i -except the universe- therefore exists as a component of another entity e . The entity e is called *environment* of entity e_i .

We express this formally in XCM ontology by asserting that, if two instances of Entity Class, e_i and e are related by **IsComponentOf** object property instance, then e is an *environment* of e_i . Therefore, we define a new class called **Environment** as equivalent to **CompoundEntity**, since every compound entity represent an environment for its components entities.

Thanks to its social laws, the environment e prescribes and determines the interactions between e_i and e , as well as between e_i and the e_j -i.e. they rule out the assembling and disassembling of e_i with the other components of e . These laws also govern the input of e_i into e , and the output of e_i from e .

Let us for example consider the case of an antenna: its environment is its coverage area, and the entering (respectively leaving) of mobile devices into (respectively from) it is controlled by its social laws. When its social laws confer to an entity the capacity to initiate operations modifying its own structure (internal autonomy) or its relations with its environment (external autonomy), this entity is commonly called an agent.

These facts are expressed in our ontology by the following elements: We define *HasLaw* the object property expressing the fact that environment e has social law L attached to its organisation; *LawOf* the object property defined as inversed *HasLaw* object property expressing that law L is related to environment e , which means that in order to interact together, the components of the environment e must exchange messages that are on conformity with the law L . Therefore, we define $L - Message$ as an object property, which relates the M messages that are on conformity with L law. We present here a partial XCM code on this concern:

```

<owl:Class rdf:ID="Environment">
    <owl:equivalentClass rdf:resource="#CompoundEntity"/>
</owl:Class>

<owl:Class rdf:ID="Law">
    <rdfs:subClassOf rdf:resource="#Entity"/>
</owl:Class>

<owl:Class rdf:ID="Message">
    <rdfs:subClassOf rdf:resource="#Entity"/>
</owl:Class>

<owl:ObjectProperty rdf:ID="HasLaw">
    <rdfs:domain rdf:resource="#Entity">
        <rdfs:range rdf:resource="#Law">
    </owl:objectProperty>

<owl:ObjectProperty rdf:ID="LawOf">
    <rdfs:domain rdf:resource="#Law">
        <rdfs:range rdf:resource="#Entity">
    </owl:objectProperty>

<owl:ObjectProperty rdf:ID="L-Message">
    <rdfs:domain rdf:resource="#Message">

```

```
<rdfs:range rdf:resource="#Law">
</owl:objectProperty>
```

An entity can be tight to several environments. However, due to the enrooting of "ubiquitous" entities in the physical space, an entity can not be physically present in two different environment in the same time. This means it can be active at the most in one environment and "virtually" or "sensorially" present in other environments (cf.Ports). To express this property, we introduce in our ontology the two object properties *IsVirtuallyPresentIn* and *IsPhysicallyPresentIn*, to express if entity is physically or virtually present in environment.

The notion of environment then encompasses within a single concept all the semantic diversity of the ubiquitous application components: a social semantics, inherited from coordination in general, and a physical semantics of entities, which becomes essential as soon as the entities are evolving onto -for example mobile- devices subjected to the laws of the physical space.

By social semantics, we mean the capacity of an entity to belong to a social structure, such as a group of entities taht it is interacting with (for instance a person belongs to a group of persons with which it is presently in meeting).

The XCM model supports multiple organisational linking of an entity (for instance a person is linked to a football club as member, but also and mainly to a company in the name of which it is predominantly acting during the meeting).

By physical semantics,we namely mean the impossibility for an agent to act in two environments at the same time, or to be "teleported" from one environment to another . An entity can however remain " aware of " another environment than the one in which it is active. As we will see further, it can open some specific communication channels in this environment, thus implementing a remote perception mechanism.

3.3 Ports

A port is a special type of entity dedicated to communication between entities. A port p has the specificity to be generally active while being coupled to an agent a_i , which is the port's master.

We define therefore in our ontology an Agent as subclass of *Entity* class, with additional restrictions on the range of **HasPort** object property. In **Agent** class, the cardinality of the property **HasPorts** must be at least equal to 1 indicating all instances of this class has ability to communicate with external world with at least one port.

The coupling between a_i and p is obtained through a special type of composition called interface, which is specified by social laws (of p and a , and of their common environment). This is expressed in our ontology by adding **InterfaceComposition** class as subclass of *Law* class, and by coupling between an individual from *Entity* class and individual from *Port* class, through **HasInterfaceComposition** objec property.

These compositional laws define how the port is assembled to its master, for example maintained versus not maintained by master's movement, linked by ownership, or by usage, etc. They may also define the modalities of using the port (in terms of communication protocol, of bandwidth, etc). For answering ubiquitous computing needs, we also distinguish removable and irremovable ports.

Example: For a human agent, a mobile phone is a removable port, whereas an audio-prosthesis is irremovable. A pair of glasses is somewhere in between, obeying to coupling laws, that are stronger than the phone's ones, but looser than the prosthesis ones. An agent a may be coupled to several ports. It can acquire ports, and dissociate itself from ports dynamically. The agent-port assembling and disassembling procedures are triggered either explicitly, by an initiative of A , or implicitly by the entrance into a new environment, or by the environment dynamics, which may for example welcome new ports, which are automatically coupled to A .

Partial XCM ontology definition about ports is presented here:

```

<owl:Class rdf:ID="Port">
  <rdfs:subClassOf rdf:resource="#Entity"/>
  <rdfs:subClassOf>
    <owl:restriction>
      <owl:OnProperty rdf:resource="#HasOwner">
        <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
        </owl:minCardinality>
      </owl:restriction>
    </rdfs:subClassOf>
  </owl:Class>

<owl:Class rdf:ID="Agent">
  <rdfs:subClassOf rdf:resource="#Entity"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:OnProperty rdf:resource="#HasPort"/>
      <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
      </owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

```

<owl:ObjectProperty rdf:ID="HasPort">
  <rdfs:domain rdf:resource="#Entity"/>
  <rdfs:range rdf:resource="#Entity"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="HasOwner">
  <rdfs:domain rdf:resource="#Port"/>
  <rdfs:range rdf:resource="#Entity"/>
</owl:ObjectProperty>

```

The notion of port is then a fundamental mechanism, which confers to XCM the ability to coordinate context-sensitive entities. This context-awareness is the central characteristics of application components in ubiquitous computing.

4 Related Works

Number of interesting frameworks are investigating pervasive computing research, such as ContextToolkit [11], Cooltown [17], Intelligent Room [16], OneWorld [15], EventHeap [8]. These systems use ad hoc representations of context knowledge, while GAIA [13], Cobra [14] and Semantic Gadgets [3], explore the use of ontology to represent context knowledge. EventHeap[8] is the first tuplespace coordination model intended to pervasive computing rooms.

Our contribution regarding theses systems is a generic approach for coordination in pervasive computing based on a global representation of context and physical world, and the use of Semantic Web languages to formalize ontologies of context, providing an explicit representation of contexts for reasoning and knowledge sharing. The proposed XCM model is used in UbiDev[19] framework and CB-Sec[18] framework.

5 Conclusion

Ontologies and explicit context representation are key requirements for realizing pervasive systems. Our works show that the newly emerged Web Ontology Language OWL is suitable for building a common knowledge representation for pervasive computing to deal with context-aware interactions.

Through some abstract concepts -entity, environment, social laws and port-XCM takes place in a layer architecture allowing to apprehend in a conceptually simple and homogeneous way the diversity and the dynamics of pervasive application universes. It integrates in particular the immersion of the application components within the physical universe, and the context-sensitivity required by the pervasive applications. XCM model makes computer's interaction closer to humans one's in term of context and environment dependency, and knowledge sharing.

References

1. M. Weiser The computer for the 21st century. *Scientific American*, 265(30):94,104, 1991.
2. Albercht Schmidt, Ubiquitous Computing- Computing in Context, PHD thesis, Computing Department, Lancaster University, UK, 2002.
3. Ora Lassila, Mark Adler: Semantic Gadgets: Ubiquitous Computing Meets the Semantic Web, in: Dieter Fensel et al (eds.): Spinning the Semantic Web, pp.363-376, MIT Press, 2003
4. Lyytinen, Kalle, and Yoo Youngjin. Issues and Challenges in Ubiquitous Computing. *Communications of the ACM* 45(12): 62-65, 2002.
5. T.W. Malone and K. Crowston. The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, 26(1):87-119, March 1994.
6. N. Carriero and D. Gelernter. Coordination Languages and Their Significance. *Communications of the ACM*, 35(2):97-107, February 1992.
7. P. Ciancarini, F. Arbab and C. Hankin: Coordination languages for parallel programming. *Parallel Computing*, 24 (7):989-1004, 1998.
8. Bradley Earl Johanson, *Application Coordination Infrastructure for Ubiquitous Computing Rooms*, PHD thesis, Stanford University 2003.
9. M. Schumacher: Objective Coordination in Multi Agent Systems Engineering. Springer Verlag. LNAI 2039, 2001. (also published as PhD Thesis, Dept of Informatics, University of Fribourg, Suisse).
10. F. Varela and H. Maturana. Autopoiesis and Cognition: The realization of the Living. Boston Studies in the Philosophy of Science in Cohen, Robert S., and Marx W. Wartofsky (eds.) , Vol. 42, Dordrecht (Holland): D. Reidel Publishing Co., 1980.
11. Daniel Salber, Anind K. Dey, and Gregory D. Abowd. The context toolkit: Aiding the development of context-enabled applications. In CHI, pages 434-441, 1999.
12. Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, May 2001.
13. Anand Ranganathan, Robert E. McGrath, Roy Campbell, and Dennis M. Mickunas. Ontologies in a pervasive computing environment. Workshop on Ontologies in Distributed Systems, IJCAI 2003, 2003.
14. Harry Chen et al., *An Ontology for Context-Aware Pervasive Computing Environments*, Workshop on Ontologies and Distributed Systems, August 2003.
15. Grimm, R., et al. A System Architecture for Pervasive Computing. in 9th ACM SIGOPS European Workshop. 2000. Kolding, Denmark.
16. Coen, M.H. A prototype intelligent environment. in Cooperative Buildings Integrating Information, Organization, and Architecture First International Workshop CoBuild'98 Proceedings. 1998. Darmstadt, Germany: Berlin, Germany : Springer-Verlag, 1998.
17. Kindberg, T., et al. People, places, things: Web presence for the real world. in Third IEEE Workshop on Mobile Computing Systems and Applications. 2000. Los Alamitos CA USA: Los Alamitos, CA, USA : IEEE Comput. Soc, 2000.
18. S. K. Mostéfaoui, A. Tafat-Bouzid and B. Hirsbrunner. Using Context Information for Service Discovery and Composition. Proceedings of the Fifth International Conference on Information Integration and Web-based Applications and Services, iiWAS'03, Jakarta, Indonesia, 15 - 17 September 2003. pp. 129-138.

19. S. Maffioletti, S.Kouadri M. and B. Hirsbrunner . Automatic Resource and Service Management for Ubiquitous Computing Environments. to Appear in Middleware Support for Pervasive Computing Workshop (at PerCom '04), PerWare '04, Orlando, Florida USA, 14 March 2004.
20. Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL Web Ontology Language Reference, w3c recommendation, 10 February 2004.

Improving Web Searching Using Descriptive Graphs

Alain Couchot

Conservatoire National des Arts et Métiers, Paris, France
Laboratoire Cedric, équipe Isid (Ingénierie des Systèmes d'Information et de Décision)
couchot-a@wanadoo.fr

Abstract. Finding a relevant piece of information on the web is a challenge. We propose in this paper to use simple ontologies and descriptive graphs. A simple ontology holds less information than a classical ontology. This offers two advantages: it is easy to obtain a consensus about a simple ontology, and the use of a simple ontology is more flexible and more easy, since it enforces fewer constraints. The simple ontologies are used for the building of descriptive graphs, which summarize the information held by a web ressource. The interoperability between simple ontologies is guaranteed by means of a global terminology. The query of the users is expressed by means of descriptive graphs. In order to answer the query of the user, we have to study the similarities between the descriptive graph of the query and the descriptive graphs of the web resources. In order to discover these similarities, we introduce the notions of precision of a concept, and of view of a descriptive graph.

1 Introduction

Finding a relevant piece of information on the web is a challenge. Millions of web sites are at the disposal of the users, but finding the site which will give the best answer the query of the user is a delicate work, which has not yet be solved in a satisfactory way. The search fulfilled by the current search engines is too unsatisfactory, because this search is mostly based on the exploitation of keywords, and is unaware of the semantics of the documents. Most of the documents retrieved by the search engines do not fit the query of the user. The semantic web [1] seems promising. The objective of the semantic web is to make the content of the web sites readable not only by the human reader, but also by the machine. At the moment, the machine cannot interpret the semantics of the natural language, even if some attempts to extract the data from a document have been proposed [5].

In order to find a solution to this problem, a semantic annotation layer has been added to the web documents. This layer is readable by the machine. But the annotation has to be readable not only by the computer of the information supplier. A consensus has to be established about the information structure contained in the annotation. An ontology layer has been introduced above the annotation layer [5, 7, 11]. The aim of the ontology is to give a formal and conceptualised representation of the knowledge.

However, the adoption of a global ontology by all the web users seems an utopian idea. The notion of domain ontology has been developed. A domain ontology formalizes the knowledge bound to a specific domain. It is incumbent upon experts to build a domain ontology.

However, the use of domain ontologies raises some problems: overlaps can happen, the management of the relationships between ontologies can be very delicate (experts have to maintain the relationships between ontologies permanently). Furthermore, the user will not know which ontology to choose in order to formulate his query (finding the good ontology can be so complex as finding a web site, since each information supplier can in theory create his own ontology).

In order to find a solution to these problems, we propose to use simple ontologies and descriptive graphs. A simple ontology holds less information than a classical ontology. This offers two advantages: it is easy to obtain a consensus about a simple ontology, and the use of a simple ontology is more flexible and more easy, since it enforces fewer constraints. The simple ontologies are used for the building of descriptive graphs, which summarize the information held by a web ressource. The interoperability between simple ontologies is guaranteed by means of a global terminology. The query of the users is expressed by means of descriptive graphs. The search problem is then equivalent to the problem of finding similarities between descriptive graphs.

The paper is organized as follows: the section 2 introduces the notions of global terminology and simple ontology, the section 3 defines the notion of descriptive graph, the section 4 studies the similarity between descriptive graphs, the section 5 presents the related works, the section 6 concludes.

2 Simple Ontology and Global Terminology

In this section, we introduce the notions of *simple ontology* and *global terminology*. A simple ontology is interesting, since it only contains a minimal number of notions. A consensus is more easy to obtain, and the use of the ontology is more flexible and more easy. We introduce then the notion of *global terminology*: a global terminology is a set of simple ontologies, sharing a set of common concepts. Users using different ontologies can so work together.

2.1 Simple Ontology

A *simple ontology* is a set C . The elements of the set are named *concepts*, and the simple ontology has the following properties:

- C has an irreflexive, antisymmetric, transitive relation, noted $<$, called *subsumption hierarchy*;
- C has an *universal concept*, noted U , that is a concept which has the following property: for each concept c of C different from U , we have: $c < U$.

Note that we do not specialize the notion of concept, in order to obtain ontologies which are more flexible.

Example. We consider an ontology O_1 which consists of the following concepts:

Car - Driver - Vehicle - House - Colour - Red - U

with $\text{Car} < \text{Vehicle}$; $\text{Driver} < U$; $\text{House} < U$; $\text{Vehicle} < U$; $\text{Colour} < U$; $\text{Red} < U$; $\text{Red} < \text{Colour}$.

O_1 is a simple ontology.

2.2 Global Terminology

We introduce now the notion of *global terminology*. A global terminology is interesting, since it is so possible for the users to share common concepts, even if they do not agree on the subsumption hierarchy or on the relevance of the concepts of an ontology. The notion of global terminology is useful in order to obtain the interoperability between simple ontologies. Users working with different ontologies will employ a common vocabulary.

We enforce that that concepts designated by the same word in two different ontologies have the same meaning.

A global terminology G is a set of simple ontologies, having the following properties:

(i) if the concept c belongs to the simple ontology O_1 of G and to the simple ontology O_2 of G , then the concept c has the same meaning in O_1 and in O_2 ;

(ii) if there is an ontology O_i of G which contains the concepts c_1 and c_2 , with $c_1 < c_2$, then, for each ontology O_j of G , containing c_1 and c_2 , we have: either c_1 and c_2 are not linked by the subsumption relationship, or we have: $c_1 < c_2$. We call this property the *subsumption preservation property*.

The set of concepts of G is the union of the concepts of the simple ontologies of G .

Example. We consider a simple ontology O_2 having the following concepts:

Car - House - Red - Colour - U

with: $\text{Car} < U$; $\text{House} < U$.

The set $O = \{O_1, O_2\}$ is a global terminology. (In the following, in order to simplify the examples, we will not make explicit the universal concept, we will assume that it is implicit.)

3 Descriptive Graphs

We introduce here the notion of *descriptive graph built with a simple ontology*. Informally, a descriptive graph is a graph destined to summarize the information held by a web resource. The nodes of the graph are concepts of the simple ontology associated to the graph.

3.1 Definition of a Descriptive Graph

Descriptive Graph. Let O be a simple ontology. A *descriptive graph* built with the simple ontology O is an oriented graph (N, V) , where N is a set of nodes and V is a set of edges linking pairs of nodes. Each node is labelled by a concept of the simple on-

tology O . A concept can label several nodes. Each node has at the most one incoming edge. We enforce that a descriptive graph holds the following property:

There is a node of the graph which has no incoming edge;

There is a node of the graph which has no outgoing edge;

The other nodes have one incoming edge and one outgoing edge.

Example. We consider the simple ontology O_3 containing the following concepts:

Piece of furniture - Table - Antique dealer - Customer - Store - Buy - At

with Table < Piece of furniture

We consider the following descriptive graph G_1 :

Customer* → Buy* → Piece of furniture* → At → Antique dealer

(The asterisk designates a significant node.)

Another example of descriptive graph G_2 is:

Antique dealer* → Buy* → Piece of furniture* → At → Antique dealer

We notice here that the same concept (Antique dealer) labels two nodes.

We precise now the notion of *subsumption graph*, which will be useful in the following, especially for the definition of the notion of *precision* of a concept.

Subsumption Graph. A subsumption graph of a simple ontology O , having a set of concepts C , is a graph such as the nodes are the concepts of the set C , and such as for each pair (c_1, c_2) of concepts of C , there is an edge from c_1 à c_2 iff:

- (i) $c_2 < c_1$;
- (ii) there is no concept a of C such as $c_2 < a$ and $a < c_1$.

Thanks to the notion of subsumption graph, we can now introduce the notion of *precision of a concept*. The more a concept is localized below in the subsumption graph, the more the concept fits a precise notion in the ontology.

Precision of a Concept. The precision of a concept c for a simple ontology is the length of the longest path from the universal concept to the concept c in the subsumption graph. (Note that there is sometimes several paths from the universal concept to the concept c , since a concept can have several father nodes in the subsumption graph).

We will use the following notation: $Prec(c)$.

Example. For the previous ontology O_3 , the precision of the concept Table is equal to 3, the precision of the concept Piece of furniture is equal to 2, the precision of Customer is equal to 2. (The length of the path is the number of nodes of the path.)

We use the notion of precision of a concept to define the notion of *precision of a descriptive graph*. This notion allows to represent the precision degree of the information held by a descriptive graph.

Precision of a Descriptive Graph. The *precision of a descriptive graph* G is the greatest precision of the concepts of the descriptive graph G . We will use the following notation: $Prec(G)$.

Example. The precisions of the graphs G_1 and G_2 built with the ontology O_3 are equal to 2. We consider the following descriptive graph G_3 :

Customer* → Buy* → Table → At → Antique dealer

The precision of this graph is 3. (The concept with the greatest precision is Table.)

In order to compare graphs built with different simple ontologies, we introduce the notion of *average precision*. For example, a concept can have a great precision in an ontology and a little precision in another ontology. In order to evaluate the precision

of the concept in the global terminology, we use the average of the precisions of the concept in the simple ontologies associated with the global terminology.

Average Precision of a Concept. Let G be a global terminology. We consider a concept c , which belongs to n simple ontologies of G : O_1, O_2, \dots, O_n . The average precision of the concept c is the average of the precisions of c in O_1, O_2, \dots and O_n .

We can now introduce the notion of *significant precision of a descriptive graph*. The *significant precision of a descriptive graph* takes into account the average precisions of the concepts of the descriptive graph in order to deduce an average precision of the information held by the descriptive graph.

Significant Precision of a Descriptive Graph. Let G be a descriptive graph. The *significant precision of the descriptive graph* G is the average of the average precisions of the concepts labelling a node of the descriptive graph G . We will use the following notation: $\text{Sign_Prec}(G)$.

3.2 Antecedent of a Concept

In order to observe the information held by a descriptive graph at a precision level less great than the level initially chosen by the graph designer, we introduce the notion of *antecedent of a concept*.

Precision k Antecedent of a Concept. Let c be a concept. The precision of c is n . Let k be a number strictly smaller than n . A *precision k antecedent* of the concept c is a concept c' having the following properties:

- (i) there is a path in the subsumption graph such as the first node of the path is c' and the last node of the path is c ;
- (ii) the precision of c' is equal to k .

Example. We consider the ontology O_4 having the following concepts:

Piece of furniture - Table - Leg - Decoration

with $\text{Leg} < \text{Table} < \text{Piece of furniture} < \text{Decoration}$

We consider the concept Leg . The concept Table is a precision 4 antecedent of Leg . The concept $\text{Piece of furniture}$ is a precision 3 antecedent of Leg . The concept Decoration is a precision 2 antecedent of Leg .

It is possible to prove that any precision k concept has at least one precision k' antecedent, for each k' strictly smaller than k .

Theorem. We consider a precision k concept c . For each integer k' smaller than k , there is at least one precision k' antecedent.

In order to prove the theorem, we need the following lemma:

Lemma. We consider the longest path L in the subsumption graph from the universal concept U to the precision k concept c . We consider a concept c_1 belonging to this path. Let k_1 be the rank of c_1 in the path (the rank of a node fits the position of the node in the path: for example, the universal concept has the rank 1). Then the precision of c_1 is equal to k_1 .

Proof of the Lemma. In order to prove the lemma, we use two steps.

First Step. We first show that the precision of c_1 is smaller than k_1 . Let us suppose that the precision of c_1 is strictly greater than k_1 . Then, we can build a path L_1 from the universal concept to the concept c_1 such as the length of L_1 is k_2 , with $k_2 > k_1$. It is then

possible to build a path from the universal concept to the concept c using the path L_1 to c_1 , then using the path L from c_1 to c . The length of this path is equal to $k_2 + (k - k_1)$. Since we have $k_2 > k_1$, we deduce that $k_2 + (k - k_1) > k$. We have then built a path from the universal concept to the concept c such as the length of the path is strictly greater than k . This is contradictory with our hypothesis (since k is the length of the longest path from the universal concept to the concept c). Then the precision of the concept c_1 cannot be strictly greater than k_1 .

Second Step. We show now that the precision of c_1 is greater than k_1 . Let us suppose that the precision of c_1 is strictly smaller than k_1 . Since it is possible to build a path from the universal concept to the concept c_1 such as the length of the path is k_1 , there is a contradiction with the fact that the precision of c_1 is strictly smaller than k_1 . We deduce that the precision of c_1 is greater than k_1 .

Since the precision of c_1 is both smaller than k_1 and greater than k_1 , it is equal to k_1 . We have proved the lemma.

Proof of the Theorem. For each integer k' strictly smaller than k , there is one concept with the rank k' in the longest path from the universal concept to the concept c . The precision of this concept is k' , according to the previous lemma. This concept is an antecedent of c , since it is in a path leading to the concept c . Consequently, there is always a precision k' antecedent for c .

From a descriptive graph G , there is a set of descriptive graphs which represent the information held by G at a precision level smaller than the precision level of G (this is due to the fact that a concept can have several father concepts in the subsumption hierarchy). In order to solve this problem, we introduce the notion of *composite concept*. Thanks to this notion, we can build an unique graph in order to represent the information held by G at a precision level smaller than the precision level of G .

Composite Concept. Let O be a simple ontology associated to the set of concepts C . A concept of C is a *simple concept*. A simple concept is also a *composite concept*. The conjunction of two composite concepts is also a composite concept.

Example. We consider the ontology O_5 containing the following concepts:

Table – Piece of furniture – Object - Goods

with Table < Piece of furniture < Object and Piece of furniture < Goods

We can build the composite concept “Object AND Goods”.

The utility of the introduction of the notion of composite concept comes from the fact that it is so possible to group all the precision k' antecedents of a precision k concept (where k' is an integer smaller than k).

Precision k' Composite Antecedent. We consider the precision k concept c and an integer k' strictly smaller than k . The *precision k' composite antecedent of the concept c* is the concept composite built with the conjunction of all the precision k' antecedents of the concept c .

From the previous theorem, we know that each precision k concept has a precision k' composite antecedent.

Algorithm for the Building of the Precision k' Composite Antecedent of the Concept c .

We pose $X = \{ c \}$ and $p = \text{Prec}(c)$.

($\text{Prec}(c)$ is the precision of the concept c).

Let z be an integer varying between 1 and $(p - 1)$. We initialize $(p - 1)$ sets $E(c, z)$ with the value \emptyset .

At the end of the procedure, each set $E(c, z)$ will contain all the precision z antecedents of c .

Procedure having the incoming parameters X and p

While $p > 1$

For each concept y such as there is an edge in the subsumption graph from the concept y to a concept of the set X :

Add the concept y to the set $E(c, \text{Prec}(y))$

Pose $X = E(c, p - 1)$

Pose $p = p - 1$

Endfor

Endwhile

At the end of the procedure, we build the precision z composite antecedent: $AC(c, z)$ using the set $E(c, z)$. If $E(c, z) = \{c_1, c_2, \dots, c_N\}$, then $AC(c, z) = c_1 \text{ AND } c_2 \text{ AND } \dots \text{ AND } c_N$.

In order to compare descriptive graphs, we need to evaluate the precision of the composite antecedents.

Average Precision of a Composite Antecedent. We consider a composite antecedent AC of a concept c . Let us suppose that AC is equal to $c_1 \text{ AND } c_2 \text{ AND } \dots \text{ AND } c_n$ (where each c_i is a simple concept). The *average precision of this composite antecedent* is equal to the average of the average precisions of the concepts c_i . (Notice that all the concepts c_i have the same precision, according to the definition of the composite antecedent, but they do not have necessarily the same average precision.)

We will need to compare descriptive graphs at any precision level, so we will need to compare composite concepts.

Partial Identity of two Composite Concepts. Let C and D be two composite concepts. Let us suppose that $C = c_1 \text{ AND } c_2 \text{ AND } \dots \text{ AND } c_n$ and $D = d_1 \text{ AND } d_2 \text{ AND } \dots \text{ AND } d_k$ (where c_i and d_j are simple concepts). We say that C and D are *partially identical* iff there is a concept c_i and a concept d_j such as $c_i = d_j$.

Example. The composite concepts “Land-vehicle AND Amphibian-vehicle” and “Land-vehicle AND Flying-vehicle” are partially identical. In the first composite concept, we are interested by a concept which is at the same time a land vehicle and an amphibian vehicle. In the second composite concept, we are interested by a concept which is at the same time a land vehicle and a flying vehicle. So the two composite concepts represent a concept which is a land vehicle, this is why we claim that these two composite concepts are partially identical.

4 Similarity Between Descriptive Graphs

In this section, we will study the similarity between descriptive graphs. We first introduce the notion of *view of a descriptive graph*, and we define the notion of *similarity coefficient*.

4.1 View of a Descriptive Graph at the Level k

Let n be the precision of a descriptive graph G . Let k be a number strictly smaller than n . We build a *view of the graph G at the level k* in the following way:

Let N be a node of the graph G labelled by the concept c . If the precision of c is smaller than k , the labelling of N is not modified. If the precision of c is strictly greater than k , then the labelling of N is replaced by the precision k composite antecedent of the concept c .

In order to designate the view of G at the level k , we will use the following notation: $V(G, k)$.

Example. We consider the descriptive graph G_3 built with the simple ontology O_3 :

Customer* → Buy* → Table → At → Antique dealer

The precision of this graph is 3. A view at the level 2 of this graph is:

Customer* → Buy* → Piece of furniture → At → Antique dealer

In order to compare descriptive graphs at different precision levels, we have to precise the notion of *identity of two views*.

Identity of two Views. Let G_1 and G_2 be two descriptive graphs. We consider two views $V(G_1, k_1)$ and $V(G_2, k_2)$. $V(G_1, k_1)$ can be written as follows: $c_1 \rightarrow c_2 \rightarrow \dots \rightarrow c_N$ and $V(G_2, k_2)$ can be written as follows: $d_1 \rightarrow d_2 \rightarrow \dots \rightarrow d_P$ (where the concepts c_i and d_j are composite concepts).

The two views are identical iff:

(i) $P = N$

(ii) for each integer i between 1 and N , the composite concepts c_i and d_i are partially identical.

4.2 Similarity Coefficient Between Two Descriptive Graphs

We consider two descriptive graphs G_1 and G_2 , respectively associated to two simple ontologies O_1 and O_2 . We build the views of G_1 at the level k (using the ontology O_1), for k varying between ($Prec(G_1)$ - 1) and 1. We also build the views of G_2 at the level k' (using the ontology O_2), for k' varying between ($Prec(G_2)$ - 1) and 1.

Let us suppose that there are two integers k_1 and k_2 such as:

(i) the view of G_1 at the level k_1 and the view of G_2 at the level k_2 are identical;

(ii) the view of G_1 at the level k_1 and the view of G_2 at the level $(k_2 + 1)$ are not identical;

(iii) the view of G_1 at the level $(k_1 + 1)$ and the view of G_2 at the level k_2 are not identical.

We say then that G_1 is *similar* to G_2 with a similarity coefficient equal to:

$$(Sign_Prec(V(G_1, k_1)) + Sign_Prec(V(G_2, k_2))) / (Sign_Prec(G_1) + Sign_Prec(G_2))$$

The similarity coefficient is a number between 0 and 1. The more the similarity coefficient is close to 1, the more the graphs are similar. The following proposition proves that there is at the most one only similarity coefficient for two graphs G_1 and G_2 .

Proposition. There is at the most one only pair (k_1, k_2) which satisfies (i), (ii) and (iii).

Proof. Let us suppose that there are two pairs (k_1, k_2) and (k'_1, k'_2) which satisfy (i), (ii) and (iii). Let us suppose without loss of generality that $k'_1 > k_1$. The greatest precision of the concepts of $V(G_1, k'_1)$ is then greater than the greatest precision of the concepts of $V(G_1, k_1)$. As the concepts of $V(G_1, k'_1)$ are the same than the concepts of $V(G_2, k'_2)$, as the concepts of $V(G_1, k_1)$ are the same than the concepts of $V(G_2, k_2)$, and as we have the subsumption preservation property (see section 3: definition of the global terminology), we deduce that the greatest precision of the concepts of $V(G_2, k'_2)$ is greater than the greatest precision of the concepts of $V(G_2, k_2)$. So, we have: $k'_2 > k_2$. All the views $V(G_1, m_1)$ and $V(G_2, m_2)$ with m_1 varying between k_1 and k'_1 , and with m_2 varying between k_2 and k'_2 , are identical. Then, in particular, $V(G_1, k_1)$ and $V(G_2, (k_2 + 1))$ are identical. This is contradictory with the property (ii). It is not possible to have two pairs (k_1, k_2) which satisfy (i), (ii) and (iii).

Example. We consider the ontology O_5 containing the following concepts:

Customer - Antique dealer - Buy - At - Piece of furniture - Table - Leg - Decoration - Good - Seller with Leg < Table < Piece of furniture < Decoration and Piece of furniture < Good and Antique dealer < Seller

We consider the following ontology O_6 :

Customer - Seller - Bibelot - Decoration - Buy - At

with Bibelot < Decoration

We suppose that the global terminology just contains the simple ontologies O_5 and O_6 .

We consider the following descriptive graph G_4 built with the simple ontology O_5 :

Customer → Buy → Leg → At → Antique dealer

We also consider the following graph G_5 built with the ontology O_6 :

Customer → Buy → Bibelot → At → Seller

The precisions of the concepts in the ontology O_5 are:

$Prec(\text{Customer}) = 2$; $Prec(\text{Antique dealer}) = 3$; $Prec(\text{Buy}) = 2$; $Prec(\text{At}) = 2$; $Prec(\text{Piece of furniture}) = 3$; $Prec(\text{Table}) = 4$; $Prec(\text{Leg}) = 5$; $Prec(\text{Decoration}) = 2$; $Prec(\text{Good}) = 2$; $Prec(\text{Seller}) = 2$.

The precisions of the concepts in the ontology O_6 are:

$Prec(\text{Customer}) = 2$; $Prec(\text{Seller}) = 2$; $Prec(\text{Bibelot}) = 3$; $Prec(\text{Decoration}) = 2$; $Prec(\text{Buy}) = 2$; $Prec(\text{At}) = 2$.

The precision of the graph G_4 is 5, and the precision of the graph G_5 is 3.

The views of the graph G_4 are:

$V(G_4 ; 4) : \text{Customer} \rightarrow \text{Buy} \rightarrow \text{Table} \rightarrow \text{At} \rightarrow \text{Antique dealer}$

$V(G_4 ; 3) : \text{Customer} \rightarrow \text{Buy} \rightarrow \text{Piece of furniture} \rightarrow \text{At} \rightarrow \text{Antique dealer}$

$V(G_4 ; 2) : \text{Customer} \rightarrow \text{Buy} \rightarrow \text{Decoration} \rightarrow \text{At} \rightarrow \text{Seller}$

The view of G_5 at level 2 is:

$V(G_5 ; 2) : \text{Customer} \rightarrow \text{Buy} \rightarrow \text{Decoration} \rightarrow \text{At} \rightarrow \text{Seller}$

The view of the graph G_4 at level 2 and the view of the graph G_5 at level 2 are identical. Consequently, the graphs G_4 and G_5 are similar with a similarity coefficient equal to $(2 + 2)/(2.8 + 2.2) = 0.8$.

(The significant precision of the graph G_4 is 2.8, the significant precision of $V(G_4, 2)$ is 2 ; the significant precision of the graph G_5 is 2.2 ; the significant precision of $V(G_5, 2)$ is 2).

4.3 Overview of Web Searching

We give now an overview of the principle of web searching. Simple ontologies built from a global terminology are at the disposal of the users and the designers of web resources to build descriptive graphs of the web resources and of the queries. When a descriptive graph is created, the views of the graph are created by the system.

In order to answer a query R , the system ranks the descriptive graphs G corresponding to the web resources using the similarity coefficient between R and G . Techniques inserted in contemporary search engines can be used to find a short-listing of the graphs G . Indeed, these techniques can be used to answer the following question: when k and k' are chosen, which are the descriptive graphs G such as the view of the graph G at the level k' is identical to the view of the query R at the level k ? (Due to space limitation, the detailed process is not given here.)

5 Previous Works

Web search engines, as described by [2], do not exploit ontologies. This limits the relevance of the results which they supply. They do not take into account the context of the search. [11] proposes to couple the semantic annotations and the traditional search engines, introducing statistical links between RDF triples and terms. [7] claims that the languages based on XML do not fit the knowledge representation, and proposes the use of knowledge representation languages such as the conceptual graphs. An ontology is used. [4] proposes a search tool for the web, Shoe, allowing to define ontologies, to annotate the documents with categories and relationships of an ontology, and search on the web. These previous works are focussed around the problem of the search inside a group of documents sharing the same data structures. Our proposal does not suppose any common structure of the documents. Furthermore, when a user wants to make a search using a ontology (which is not a simple ontology such as we have defined it), he has to know and understand the concepts, the relations between the concepts, and he has to agree with them, which seems not possible in the context of the web. In our framework, the user has just to know a global terminology, and choose a hierarchy. (Note that, if he does not agree with one of the proposed hierarchies, he can build his own hierarchy.)

Some works focus on using several ontologies [10]. In addition to the problems mentioned above, this raises new difficulties [6]: for example, there may exist overlaps between ontologies, the same concepts may not have the same scope in different ontologies. Furthermore, it is very difficult for the user to manipulate several ontologies (Which ontology to choose? Has the user to know and understand all the ontologies?) In our framework, these problems do not exist, since a global terminology is used.

In the field of the comparison of graphs, [8, 9] proposes a method to compare two conceptual graphs, in the context of the representation of texts under the form of conceptual graphs. [8, 9] takes into account, for the similarity measure, a generalization hierarchy provided by the user. This method cannot be applied in our context, since

the same generalization hierarchy has to be used for the two conceptual graphs for [8, 9]. In our context, we study the similarity of graphs such as the concepts belong to different generalization hierarchies. Furthermore, the algorithm proposed by [8, 9] needs to compare each graphs pair to decide about the similarity of the graphs: in the context of the web, this would need to compare millions of graphs pairs for each query. Thanks to the concept of view that we have introduced, the pertinent resources can be short-listed using the techniques inserted in the contemporary search engines.

6 Conclusion

In order to use the web resources, a consensus seems stand out for the semantic annotations of the web resources and for the use of ontologies. We have proposed in this paper to annotate the web resources with descriptive graphs built with simple ontologies. The use of simple ontologies is flexible and easy, since no constraints about the concepts are imposed.

A descriptive graph is intended to summarize the information contained in the web resources. The user specifies his query using a descriptive graph. In order to answer the query of the user, we have to study the similarities between the descriptive graph of the query and the descriptive graphs of the web resources. In order to discover these similarities, we have introduced the notions of precision of a concept, and of view of a descriptive graph. Thanks to the view of a descriptive graph, it is possible to compare the descriptive graphs at several precision levels. So, a more precise information retrieval on the web can be achieved.

In the future, we plan to elaborate a tool for the automatic building of the descriptive graph associated to a resource. We also plan to propose a translation tool, in order to allow the formulating of the queries using the natural language, the translation tool will choose the best ontology and will generate the corresponding descriptive graph.

References

1. T. Berners-Lee, J. Hendler, O. Lassila. The Semantic Web: A New form of Web Content that is Meaningful to Computers Will Unleash a Revolution of New Possibilities, *Scientific American*, May 2001.
2. S. Brin, L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proc. of the 7th Int. World Wide Web Conf.*, Brisbane, Australia, April 1998.
3. S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingram, T. Kanungo, S. Rajagopalan, A. Tomkins, J.A. Tomlin, J.Y. Zien, Sem Tag, Seeker. Bootstrapping the Semantic Web via Autamated Semantic Annotations. In *Proc. of the 12th Int. World Wide Web Conference*, Budapest, Hungary, May 2003.
4. J. Heflin, J. Hendler. Dynamic Ontologies on the Web. In *Proc. of the 17th Nat. Conf. on Artificial Intelligence and 12th Conf. on Innovative Applications of Artificial Intelligence*, Austin, Texas, USA, August 2000.

5. B. Hammond, A. Sheth, K. Kochut. Semantic Enhancement Engine: A Modular Document Enhancement Platform for Semantic Applications over Heterogeneous Content. In *Real World Semantic Web Applications*, V. Kashyap et L. Shklar ed., IOS Press, pages 29-49, December 2002.
6. M. Klein. Combining and Relating Ontologies: an Analysis of Problems and Solutions. In *Proc. of the Workshop on Ontologies and Information Sharing*, August 2001, Seattle, USA.
7. P. Martin, P. Eklund. Embedding Knowledge in Web Documents. In *Computer Networks*, vol. 31, 11-16, pp 1403-1419, May 1999.
8. M. Montes-y-Gomez, A. Lopez-Lopez, A. Gelbukh. Flexible Comparaison of Conceptual Graphs. In *Proc. of the 11th Int Conf on Database and Expert Systems Applications*, London, United Kingdom, September 2000.
9. M. Montes-y-Gomez, A. Gelbukh, A. Lopez-Lopez, R. Baeza-Yates. Flexible Comparaison of Conceptual Graphs. In *Proc. of the 12th Int Conf on Database and Expert Systems Applications*, Munich, Germany, September 2001.
10. Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure. In *Proc. of the 1st Int. Conf. On Formal Ontologies in Information Systems*, Trento, Italy, June 1998.
11. U. Shah, T. Finin, A. Joshi, R.S. Cost, J. Mayfield. Information Retrieval on the Semantic Web. In *Proc. of the 10th Int. Conf. on Information and Knowledge Management*, Mc Lean, VA, USA, November 2002.

An Unsupervised WSD Algorithm for a NLP System

Iulia Nica^{1,2}, Andrés Montoyo³, Sonia Vázquez³, and M^a Antònia Martí¹

¹ CLiC - Centre de Llenguatge i Computació

Department of General Linguistics

University of Barcelona, Spain

iulia@clic.fil.ub.es, amarti@ub.edu

² Department of General Linguistics

University of Iasi, Romania

³ Research Group of Language Processing and Information Systems

Department of Software and Computing Systems

University of Alicante, Spain

{montoyo, svazquez}@dlsi.ua.es

Abstract. The increasing flow of information requires advanced free text filtering. An important part of this task consists in eliminating word occurrences with an inappropriate sense, which corresponds to a Word Sense Disambiguation operation. In this paper we propose a completely automatic WSD method for Spanish - restricted to nouns - to be used as a module in a Natural Language Processing system for unlimited text. We call it the Commutative Test. This method exploits an adaptation of EuroWordNet, Sense Discriminators, that implicitly keeps all lexical-semantic relations of its nominal hierarchy. The only requirement is the availability of a large corpus and a part-of-speech tagger, without any need of previous sense-tagging. An evaluation of the method has been done on the Senseval test corpus. The method can be easily adapted to other languages that dispose of a corpus, a WordNet component and a part-of-speech tagger.

1 Introduction

The information revolution supposes that telecommunications and information systems deal with a huge, constantly increasing volume of raw data. This requires the advance of data presentation towards formats able to allow the users access to data in a very natural way. Language technologies based on Natural Language Processing (NLP) techniques are indispensable in this evolution, thus they are crucial for the success of information systems.

Systems for NLP require abundant knowledge on language. A great difficulty in processing natural language is the ambiguity at all its levels: phonological, morphological, syntactic, semantic or pragmatic. One of the main objectives in designing any NLP system, therefore, is the resolution of ambiguity. Each type of ambiguity requires a specific resolution procedure.

In this paper we address the resolution of a particular type of lexical ambiguity, the one between the different senses a word might have in a text. This specific task is commonly referred to as Word Sense Disambiguation (WSD). The sense identifica-

tion is an “intermediate task” [20], necessary for certain NLP applications, such as Machine Translation (MT), Information Retrieval (IR), Text Processing, Grammatical Analysis, Information Extraction (IE), hypertext navigation, etc.

This association of a word occurrence to one specific word sense is achieved by accessing two different information sources, known as “context” and “external knowledge sources”. The “context” of an occurrence to be disambiguated is defined as a sequence of the text around it, of variable size, that provides general and linguistic information about the text and about the occurrence: syntactic relationships, semantic categories, distance, etc. “External knowledge sources” include structured lexical resources, with general knowledge associated to word senses (dictionaries, thesauri, lexical data bases), or sense-tagged corpora, where particular knowledge on senses uses can be extracted from.

The approaches to WSD are usually divided into knowledge-driven and data-driven. An important part of the research in the field of WSD was done from the knowledge-driven approach. This WSD systems exploit MRDs, thesauri or lexical databases as WordNet, frequently combined with corpora. So, Lesk [9] proposes a method that counts the overlaps of content words between each dictionary definition for word senses in Longman’s Dictionary of Contemporary English (LDOCE) and the context of the ambiguous word occurrences. Cowie [3] optimises the Lesk’s method by means of simulated annealing technique, disambiguating simultaneously all the words in the sentence and reducing thus the combinatorial explosion. In another optimisation of Lesk’s method, Wilks [24] uses co-occurrence data in the definitions from LDOCE to construct word-context vectors and, from here, word-sense vectors. Yarowsky [22] derives different classes of words from a corpus starting with the common categories of words in Roget’s Thesaurus, by considering the most frequently cooccurring words with the units from each thesaurus category. Relying on the hypothesis that each sense of a given word pertains to a different thesaurus category, he then counts the more frequent category in the context of the ambiguous occurrence. Voorhees [19] derives a construction called *hood* that represents different sense categories in the WordNet noun hierarchy. Sussna [18] defines a meter-based measurement for the semantic distance between the nouns in the text. It assigns weights to WordNet links, on the basis of type of relation they express (synonymy, hyperonymy, etc.), and it counts the number of arcs of the same type leaving a node, as well as the total depth of the arcs. He applies this metric on the shortest path between the nodes corresponding to the senses of the words in the text, in order to identify the senses at the minimal semantic distance. In order to avoid the irregular distances between the nodes in the WordNet hierarchy, Resnik [14] computes the commonly shared information content of words. It is a measure for the specificity of the concept that subsumes the words included in the WordNet IS-A hierarchy: the more specific this parent concept, the more related the subsumed words. Resnik [16] presents a measure of semantic similarity in an IS-A taxonomy, based on the notion of commonly-shared information content. Agirre [1] exploits the notion of conceptual density, a complex measure of the distance among different concepts in the WordNet noun taxonomy. Independently on the number of nouns that intervene in the computation, the metric takes into account the following parameters: the length of the shortest path between the concepts; the deepness in the hierarchy; the density of concepts in a delimited area of the hierarchy. Rigau [17] combines a set of unsupervised algorithms that can accurately disambiguate word senses in a large, completely untagged corpus. Hale [5] measures the semantic similarity on a combination of *Roget’s Inter-*

national Thesaurus and WordNet considered as taxonomies by using four metrics. Leacock [8] finds examples for the different senses of a word by using the monosemous words in synsets related to these senses in WordNet. Mihalcea [10] tries to overcome these limitations with the help of other information, as the glosses in WordNet, and by substituting the corpus with Internet. As this brief bibliographical overview makes evident, WordNet and its multilingual variant, EuroWordNet, practically have become the standard lexical devices in WSD.

The method we propose is a knowledge-driven one, as it uses information from a structured lexical source derived from EuroWordNet, and unsupervised, as it does not need a sense-tagged training corpus¹. It requires only a large corpus, a minimal pre-processing phase (POS-tagging) and very little grammatical knowledge, so it can easily be adapted to other languages that dispose of a corpus, a WN component and a part-of-speech tagger. The particularity of our WSD system is that sense assignment is performed using also paradigmatic information extracted from corpus. Thus it makes an intensive use of sense untagged corpora for the disambiguation process.

The paper is organised in the following way: we first present the general architecture of the NLP system (section 2), then the method of WSD (section 3), the evaluation process (section 4) and finally the conclusions and future work (section 5).

2 Architecture for the PLN System with the WSD Module

The WSD method we propose is to be integrated as a module into a NLP system meant to prepare a input text for a specific application: IR, IE, MT, etc. This preprocessing phase consists in annotating the text with linguistic information, in order to improve the results of the final task. The architecture of the system follows the principal levels of linguistic analysis for a text: morphological, syntactic and semantic. The preprocessing starts with the part-of-speech tagging of the input text, which consists in two operations: first, the identification of all possible POS-tags for the lexical units, with the help of the analyser MACO [2]; second, the disambiguation, which selects a unique morphosyntactic category from the possible ones, using the tagger RELAX [2]. Next, the shallow parsing is performed with the help of a grammar that identifies the sentence's constituents (np, pp, p, verbal chunks). The third step is performed by the WSD module, which consults the lexical source called “Sense Discriminators”, derived from EuroWordNet (section 3.3), and assigns a sense to the nouns from the input text. The output text of this preprocess will have annotated all this information: POS-tags, chunks and noun senses. The process is illustrated in Figure 1.

¹ This meaning of the term *unsupervised* was ultimately imposed in WSD by the two Senseval exercises [6], [4], and the term was enlarged to include also the knowledge-based WSD methods. It moves away from the classical meaning it has in Artificial Intelligence, where it denotes Machine Learning methods whose training is done without preestablished classification categories.

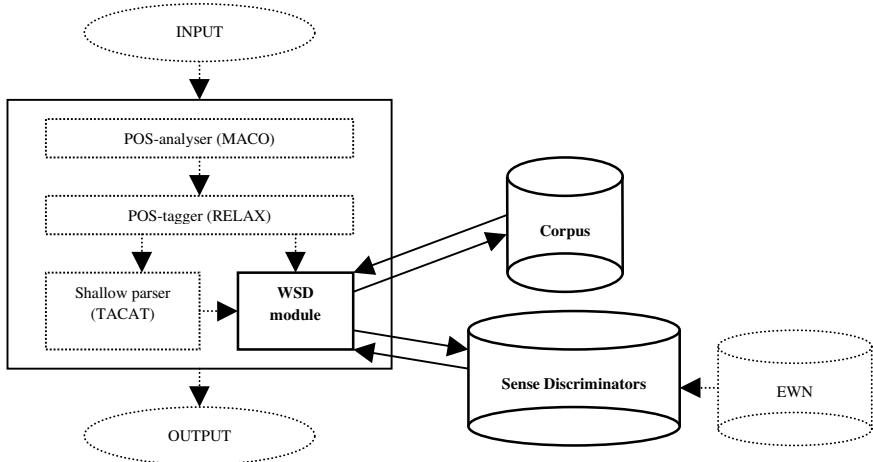


Fig. 1. NLP system with WSD module

3 WSD Method²

We propose a method for sense assignment which starts from the present “gap” in WSD between the available lexical sources and the text to be disambiguated: the lexicons provide principally paradigmatic information, meanwhile the context of the occurrences to be disambiguated offers rather syntagmatic information [7].

This limits the efficient use of the paradigmatic information from lexical sources and so the WSD process quality. The actual proposals to solve the problem are limited by the need of human intervention. The data-driven approach tries to extract syntagmatic information related to senses from manually sense-tagged corpora, as the automatic construction of such corpora is not yet efficient enough. The knowledge-driven approach, by the other hand, is centred on the approximation of corpus to lexicon, that is on the enrichment of the lexicons with manually acquired syntagmatic information from corpora. In both cases, WSD systems work with syntagmatic information.

Our basic idea, in order to reduce the gap problem, is to reverse the disambiguation process: operate on paradigmatic instead of syntagmatic information. To do it, we extract paradigmatic information associated to a given ambiguous occurrence and then map it to the paradigmatic information from the lexicon. At the base of our approach it lays the interaction that holds in natural language between the syntagmatic and paradigmatic axes: semantically similar words can substitute each other in the same context and, inversely, words that can commute in a context have a good probability to be close semantically.

² We have introduced elements of this approach to WSD in our previous works [12],[13].

For the formal treatment of the context from this perspective, we introduce the term of syntactic pattern: a triplet X-R-Y, formed by two lexical content units X and Y (nouns, adjectives, verbs, adverbs) and a relational element R, which corresponds to a syntactic relation between X and Y. Intuitively, a syntactic pattern is a sequence of words whose POS-tags combinations can satisfy a syntactic relation. Examples: [*grano*-noun *de*-preposition *azúcar*-noun], [*pasaje*-noun *subterráneo*-adjective]. In order to identify and exploit the syntactic patterns of an ambiguous occurrence, we first define a list of basic patterns, in terms of parts-of-speech (POS), that covers the possible syntactic relations involving nouns. We complement this pattern with a list of possible realisations of the basic patterns and with a set of decomposition rules to delimit the pure patterns from these realisations, such as to allow the patterns identification both in the context and in the search corpus.

The integration of the ambiguous occurrence in a syntactic pattern allows us to identify, into the corpus, information of paradigmatic type associated to the occurrence. The final information we collect for the ambiguous occurrence is the one extracted from corpus by using all its syntactic patterns and the one provided by the sentential context.

The sense assignment corresponds then to the mapping of this information to the information provided for the word senses in EWN. As word senses in EWN are defined by the lexical-semantic relations, this means we are looking for lexical-semantic relations from EWN between the word to be disambiguated and the others inside every set previously obtained for its ambiguous occurrence. We do it by means of the Commutative Test, which makes use of Sense Discriminators, an own adaptation of EWN. This new lexical source provides discrimination information for sense identification.

We detail below all these steps as well as the Sense Discriminators device derived from EWN and the WSD algorithm, the Commutative Test.

3.1 Syntactic Patterns Identification

In order to identify occurrences for a noun in EWN and their syntactic patterns, we work on a POS-tagged corpus: the corpus of the EFE News Agency (over 70 millions words), annotated with the POS-analyser MACO and the POS-tagger RELAX for Spanish (cf. section 2). The identification of the syntactic patterns is done with the help of a list of predefined patterns of morphosyntactic categories, following the structural criterion: we consider only the sequences corresponding to one of these patterns.

The delimitation of the syntactic patterns that contain the ambiguous occurrence is performed both at the lemmas and morphosyntactic categories levels. To do this, the sentence is previously POS-tagged, with the help of the same devices: MS-ANALYZER and RELAX. The sense assignment is very dependent on the quality of the pattern identification; in order to assure a proper extraction of syntactic patterns, we preferred to limit it in a first step to a part of the structural variants in which a noun can appear. Therefore, on the basis of the distributional and syntactic properties

of nouns, we predefined the following basic types of syntactic patterns for nouns in terms of morphosyntactic categories³:

N1, N2;
 N1 CONJ* N2;
 N1 PREP N2,
 N ADJ;
 N PART;
 ADJ N;
 PART N,

where we limited the conjunctions to a few coordinative ones: CONJ* = {y, o, e, u}. For the disambiguation of a particular noun *lemma0*, we define specific basic patterns. So, we particularise the general basic pattern [N1, N2] to the patterns [*lemma0*-N, N] and [N, *lemma0*-N], and [N1 CONJ* N2] to [*lemma0*-N CONJ* N] and [N CONJ* *lemma0*-N], as the noun to be disambiguated can occupy both nominal positions.

The patterns can have discontinuous realisations in texts. To cover these cases, we pre-establish the following morphosyntactic schemes for the search into the corpus orientated to the identification of the patterns, where we consider facultative elements as ADJ, ADV or DET. So, the search scheme corresponding to the particular patterns: [*lemma0*-N, N] is [*lemma0*-N ((ADV) ADJ/PART), (DET) N] and the ones corresponding to the pattern [*lemma0*-N CONJ* N] are [*lemma0*-N ((ADV) ADJ/PART) CONJ* (DET) N] and [*lemma0*-N CONJ* (DET) ((ADV) ADJ/PART) N]⁴.

The sequences identified in the corpus starting from these search schemes are to be split up into simple syntactic patterns, so we also define a set of rules to do it. For example, the scheme [*lemma0*-N ADJ1 CONJ* ADJ2] splits up into two simple syntactic patterns: [*lemma0*-N ADJ1] and [*lemma0*-N ADJ2]. So, a particular sequence as: [*coronas-corona*-N *danesas-danés*-ADJ y-y-CONJ *suecas-sueco*-ADJ], should be finally divided into: [*coronas-corona*-N *danesas-danés*-ADJ] and [*coronas-corona*-N *suecas-sueco*-ADJ].

Once we have found the sequences that correspond to the basic syntactic patterns at the morphosyntactic level, the syntactic patterns are defined at both lemma and morphosyntactic categories. For example, the pattern [*lemma0*-N, N] becomes [*lemma0*-N, *lemma1*-N], and [*lemma0*-N CONJ* N] becomes [*lemma0*-N *lemma1*-CONJ* *lemma2*-N].

3.2 Extraction of Information Related to the Occurrence

For an ambiguous occurrence of a noun appearing in the syntactic patterns Pk, we extract the following two sets:

- S1 is the reunion of the paradigms corresponding to the position of the ambiguous occurrence into each syntactic pattern Pk. Each paradigm is obtained by fixing a

³ We use the following transparent abbreviations for the morphosyntactic categories: ADJ for adjectives, ADV for adverbs, CONJ for conjunctions, DET for determinants, N for nouns, PART for past participles, PREP for prepositions.

⁴ The units between brackets are optional and the ones separated by a bare are alternatives for a position of the pattern.

given syntactic pattern at lemma and morphosyntactic levels, and letting variable only the position of the ambiguous word at lemma level.

- S2 is the set of all nouns in the sentence of the ambiguous occurrence.

3.3 Adaptation of EWN: Sense Discriminators

The adaptation of EWN is derived in the following way: for every sense X_i of a given word X in EWN, we extract the set SD $_i$ of nouns related to it in EWN along the different lexical-semantic relations. We eliminate then the common elements (at lemma level), obtaining so disjunctive sets SD $_i$. As the elements of the set SD $_i$ are related exclusively with the sense X_i , they become sense discriminators for X_i . For this reason, we called the obtained lexical device “Sense Discriminators”.

This new derivation of EWN differs from the original lexicon in the following aspects: its structure is flat, as a lexicon where each word sense has a groups of words associated to it; there is no more connection between the various senses of a word, and they have a disjunctive characterisation; the sense characterisation is done by implicit relations with words and not by explicit lexical-semantic relations with word senses.

3.4 WSD Algorithm: Commutative Test

At the basis of the algorithm it lays the hypothesis that if two words can commute in a given context, they have a good probability to be semantically close. In terms of our adaptation of EWN and of our approximation to local context, this means that if an ambiguous occurrence can be substituted in its syntactic patterns by a sense discriminator, then it can have the sense corresponding to that sense discriminator. We call this algorithm the Commutative Test (CT). In order to reduce the computational cost of this substitution operation, we perform an equivalent process: We previously extract, from corpus, the possible substitutes of the ambiguous occurrence in a syntagmatic pattern, and then we intersect this set with every set of sense discriminators; the senses for which the intersection is not empty can be assigned to the occurrence. The Commutative Test can be performed in such a straightforward way because of the sense characterisation in our adaptation of EWN is done by means of words and not by means of word senses. In fact, the algorithm operates with words from a sense-untagged corpus. Therefore, it is independent of any kind of preprocessing at sense level.

It applies on a set S of nouns related to the ambiguous occurrence in the given syntactic pattern. The set differs from one heuristic to another. The algorithm intersects S with every set SD $_i$; if it obtains a not empty intersection between S and SD $_{i0}$, then it concludes that X can have the sense X $_{i0}$ in the starting syntactic pattern.

We illustrate graphically how the algorithm works in figure 2 .

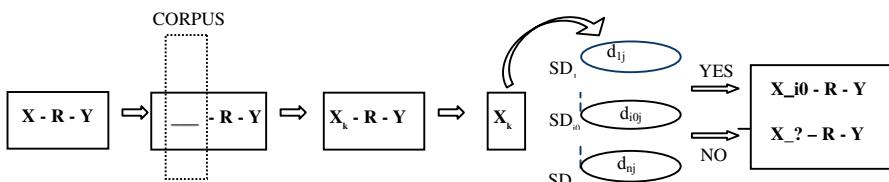


Fig. 2. The Commutative Test algorithm

3.5 WSD Heuristics. Sense Assignment

We use a WSD module that incorporates two heuristics, H1 and H2, as votants for the sense assignment. The heuristics share the Commutative Test algorithm, but they differ in the set on which the algorithm applies, S1 and S2 (section 3.2) respectively:

-H1 intersects S1 with SDi. If, for an i0, $S2 \cap SDi0 \neq \emptyset$, the heuristics concludes that X0 can have the sense i0.

-H2 intersects S2 with SDi. If, for an i0, $S2 \cap SDi0 \neq \emptyset$, the heuristics concludes that X0 can have the sense i0.

We keep all the proposed senses for the ambiguous occurrence by the two heuristics. So we keep all the possible senses the occurrence might have in the given context according to our system. We do it because the purpose of the WSD module is a semantic filtering, so we do not want to loose the correct sense.

3.6 Example

We illustrate the method for noun *órgano* in the following example from Senseval-2 (occurrence number 35):

Los enormes y continuados progresos científicos y técnicos de la Medicina actual han logrado hacer descender espectacularmente la mortalidad infantil, erradicar multitud de enfermedades hasta hace poco mortales, sustituir mediante trasplante o implantación <head>órganos</head> dañados o partes del cuerpo inutilizadas y alargar las expectativas de vida.

The steps of the disambiguation process are:

1º. *Preprocessing: input text POS-tagging*

2º. *Syntactic patterns identification for the ambiguous occurrence:*

2a. Using the search schemes, we find the following sequence: [*órganos*-N *dañados*-ADJ *o*-CONJ *partes*-N].

2b. Using the decomposition rules, we extract two basic patterns: [*órgano*-N *o*-C *parte*-N] and [*órgano*-N *dañado*-ADJ].

3º. *Extraction of information associated to the ambiguous occurrence:*

3a. From corpus, we extract the paradigm corresponding to the position of *órgano* in each of the two syntactic patterns previously identified. To do this, we let vary, at lemma level, the position of *órgano* in the two patterns: [X-N *o*-C *parte*-N] and [X-N *dañado*-ADJ] respectively. With the help of the search schemes, we then look in the corpus for the possible nouns as X in any of the possible realisations of these two patterns. We obtain two sets whose reunion is the following:

$S1 = \{mediador, terreno, chófer, árbol, cabeza, planeta, parte, incremento, totalidad, guerrilla, programa, mitad, país, temporada, artículo, tercio\}$

3b. From the context, we extract the nouns of the sentence, excepting *órgano*:

$S2 = \{progreso, científico, mortalidad, multitud, enfermedad, mortal, trasplante, implantación, órgano, parte, cuerpo, expectativa, vida\}$

4º. (In an all-nouns disambiguation task, it can be seen as an external and previous process of the derivation of the Sense Discriminators for all nouns in EWN hierarchy)
Extraction of Sense Discriminators sets

In EWN, *órgano* has five senses⁵:

- órgano_1: 'part of a plant';
- órgano_2: 'governmental agency, instrument';
- órgano_3: 'functional part of an animal';
- órgano_4: 'musical instrument'
- órgano_5: 'newspaper'.

Correspondingly, we obtain from the EWN hierarchy the following Sense Discriminators sets:

SD1: {*órgano vegetal, espora, flor, pera, manzana, bellota, hinojo, semilla, poro, píleo, carpóforo, ...*}

SD2: {*agencia, unidad administrativa, banco central, servicio secreto, seguridad social, F.B.I., ...*}

SD3: {*parte del cuerpo, trozo, músculo, riñón, oreja, ojo, glándula, lóbulo, tórax, dedo, articulación, rasgo, facción, ...*}

SD4: {*instrumento de viento, instrumento musical, mecanismo, aparato, teclado, pedal, corneta, ocarina, zampoña, ...*}

SD5: {*periódico, publicación, medio de comunicación, método, serie, serial, número, ejemplar, ...*}

5º. *Commutative Test application. WSD heuristics:*

H1: $S1 \cap SD1 = \emptyset$; $S1 \cap SD2 = \emptyset$; $S1 \cap SD3 \neq \emptyset$; $S1 \cap SD4 = \emptyset$; $S1 \cap SD5 = \emptyset$. Thus, the heuristics concludes that X0 can have the sense 3.

H2: $S2 \cap SD1 = \emptyset$; $S2 \cap SD2 = \emptyset$; $S2 \cap SD3 \neq \emptyset$; $S2 \cap SD4 = \emptyset$; $S2 \cap SD5 = \emptyset$. Thus, the heuristics also concludes that X0 can have the sense 3.

6º. *Final sense assignment:*

In this case, we obtained the same sense 3 from both heuristics, so we assign sense 3 from EWN to the occurrence of *órgano*, which corresponds to sense 2 in the Senseval-2 dictionary.

4 Evaluation

We tested this method in the Senseval-2 conditions, on all 17 words for Spanish. The results we obtained are presented in table 1:

Table 1. Results of our WSD system

	Precision	Recall	Coverage
H1	0,54166667	0,11636289	0,21483376
H2	0,59677419	0,04731458	0,07928389
H1 + H2	0,56132075	0,15217391	0,27109974

Comparative to the level reached in the Spanish Senseval-2 (precision of 51,4%-71,2%, recall of 50,3%-71,2%, coverage of 98%-100%), our method obtains low

⁵ Due to the absence of glosses in the Spanish EWN, we have created these pseudodefinitions to support our presentation, in the context of this paper.

results in terms of recall: 15%; the principal reason is its limited coverage (27%). The precision, however, is much better, with a medium of 56%, but it arrives at 100% in three cases and at 50% or more in nine cases.

We have to stress that the level of disambiguation, in H1 and thus in the combination of the two heuristics, is highly affected both by the quantity and the quality of the syntactic patterns identification. In this experiment, we have identified syntactic patterns for only 70% of the occurrences to be disambiguated, as we have considered only a part of the possible structural patterns. As we haven't used any qualitative filter on these patterns, we have obtained coverage with answers for only a 29% of them. For this reason, we do believe that the real potential of our method is higher and so the improvement of patterns delimitation is a stringent necessity.

Our principal purpose in this experiment was to verify the usefulness of the paradigmatic information for the WSD process based on a lexical source as EWN. The comparison between the two heuristics, in terms of coverage, indicates that indeed the paradigmatic information (in H1) related to an ambiguous occurrence meets more often than the syntagmatic one (in H2) with the paradigmatic information from EWN. The use of EWN as lexical source for WSD does take benefit from the paradigmatic information. Comparing H1 and H1+H2 shows us that the use of paradigmatic information improves radically the performance of the WSD algorithm: the coverage increases from 7% to 27%, with only 3% of reduction in precision, and so the recall goes up from 4% to 15%. We have thus a confirmation of our strategy to incorporate paradigmatic information from corpus in the WSD process. In terms of precision, the performance of the two heuristics is practically the same. This suggests that the two types of information, paradigmatic (in H1) and syntagmatic (in H2), are equally useful for sense assignment, thus it is necessary to exploit them both in WSD tasks.

5 Conclusions and Further Work

We propose in this paper a knowledge-driven and unsupervised method for WSD to be used as a module in a NLP system meant to prepare an input text for a real application. An important characteristic of our WSD method is its independence of any corpus tagging at syntactic or semantic level. It only requires a minimal preprocessing phase (POS-tagging) of the input text and of the search corpus, and very little grammatical knowledge.

As future work, we are studying different possibilities to improve the quality of the WSD process. So, our first interest line is the syntactic patterns identification, as a preliminar step towards disambiguation. We are then investigating the use of some other WSD algorithm(s) to apply on the information associated to the ambiguous occurrence (sets S1 and S2). In order to obtain not only a better precision but also a better coverage, we are analysing the combination of our method with some data-driven WSD method(s). Finally, we are working on an automatic partial autoevaluation of the method, for the occurrences that participate in two syntactic patterns.

References

1. Agirre, E., Rigau, G.: Word Sense Disambiguation using Conceptual Density. *Proceedings of the 16th International Conference on COLING*, Copenhagen (1996)
2. Civit, M.: *Criterios de etiquetación y desambiguación morfosintáctica de corpus en español*, Ph.D. dissertation, Universidad de Barcelona (2003)
3. Cowie, J., Guthrie, J., Guthrie, L.: Lexical disambiguation using simulated annealing. *Proceedings of the DARPA Workshop on Speech and Natural Language*, New York (1992)
4. Edmonds, P., Cotton, S. (eds.): *SENSEVAL-2: Overview. Proceedings of 2nd International Workshop "Evaluating Word Sense Disambiguation Systems"*, Toulouse (2001)
5. Hale, M.L.: Ms. A comparison of WordNet and Roget's taxonomy for measuring semantic similarity.
6. Ide, N., Véronis, J.: Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, 24(1) (1998)
7. Kilgariff, A.: Bridging the gap between lexicon and corpus: convergence of formalisms. In: *Proceedings of LREC'1998*, Granada (1998)
8. Leacock, C., Chodorow, M., Miller, G.A.: Using Corpus Statistics and WordNet Relations for Sense Identification. *Computational Linguistics. Special Issue on Word Sense Disambiguation*, 24(1) (1998)
9. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone, *Proceedings of the 1986 SIGDOC Conference*, ACM, New York (1986)
10. Mihalcea, R., Moldovan, D.: A Method for word sense disambiguation of unrestricted text, *Proceedings of the 37th Annual Meeting of the ACL*, Maryland, USA (1999)
11. Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K. J.: WordNet: An on-line lexical database, *International Journal of Lexicography*, 3(4) (1990)
12. Nica, I., Martí, M.A., Montoyo, A.: Colaboración entre información paradigmática y sintagmática en la Desambiguación Semántica Automática, *XIX SEPLN Conference*, Alcalá de Henares-Madrid (2003)
13. Nica, I., Martí, M.A., Montoyo, A., Vázquez, S.: Combining EWN and sense untagged corpora for WSD, *Proceedings of the CICLING'04*, Korea (2004)
14. Resnik, P.: Disambiguating noun groupings with respect to WordNet senses, *Proceedings of the Third Workshop on Very Large Corpora*, Cambridge (1995)
15. Resnik, P., Yarowsky, D.: A perspective on word sense disambiguation methods and their evaluation, *Proceedings of the ACL Siglex Wordshop on Tagging Text with Lexical Semantics, why, what and how?*, Washington (1997)
16. Resnik, P.: Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language, *Journal of Artificial Intelligence Research*, 11 (1999)
17. Rigau, G., Atserias, J., Agirre, E.: Combining Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation, *Proceedings of the 35th Annual Meeting of the ACL*, Madrid (1997)
18. Sussna, M.: Word sense disambiguation for free-text indexing using a massive semantic network, *Proceedings of the Second International CIKM*, Arlington, VA (1993)
19. Voorhees, E.: Using WordNet to disambiguate word senses for text retrieval, *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pittsburgh, PA (1993)
20. Wilks, Y., Fass, D., Guo, C., McDonal, J., Plate, T., Slator, B.: Providing Machine Tractable Dictionary Tools, in Pustejovsky, J. (ed.), *Semantics and the Lexicon*, Dordrecht, Kluwer (1993)
21. Wilks, Y., Stevenson, M.: *The grammar of sense: Is word sense tagging much more than part-of-speech tagging?*, Technical Report CS-96-05, University of Sheffield (1996)
22. Yarowsky, D.: Word Sense disambiguation using statistical models of Roget's categories trained on large corpora, *Proceedings of the 14th COLING*, Nantes (1992)

Enhanced Email Classification Based on Feature Space Enriching

Yunming Ye¹, Fanyuan Ma¹, Hongqiang Rong², and Joshua Huang²

¹ Department of Computer Science and Engineering, Shanghai Jiao Tong University,
Shanghai 20030, China
{yyym, fyma}@sjtu.edu.cn

² E-Business Technology Institute, The University of Hong Kong, Hong Kong, China
{hrong, jhuang}@eti.hku.hk

Abstract. Email classification is challenging due to its sparse and noisy feature space. To address this problem, a novel feature space enriching (FSE) technique based on two semantic knowledge bases is proposed in this paper. The basic idea of FSE is to select the related semantic features that will increase the global information for learning algorithms from the semantic knowledge bases, and use them to enrich the original sparse feature space. The resulting feature space of FSE can provide semantic-richer features for classification algorithms to learn improved classifiers. Naive Bayes and support vector machine are selected as the classification algorithms. Experiments on a bilingual enterprise email dataset have shown that: (1) the FSE technique can improve the email classification accuracy, especially for the sparse classes, (2) the SVM classifier benefits more from FSE than the naive Bayes classifier, (3) with the support of domain knowledge, the FSE technique can be more effective.

Keywords: Email Classification, Feature Space Enriching, Semantic Knowledge Base

1 Introduction

Email is one of the most important communication tools in everyday life. However, while it provides a convenient tool for communication, the large volume of emails also brings about the problem of 'email overload' [1] that takes users much time to process them and even that users can't efficiently and timely manage their mail repositories. Especially, the problem becomes more challenging for large organizations, such as e-government and commercial enterprises of which

¹ This paper has been supported by China NSF project (No.60221120145) and Shanghai Science and Technology Foundation (under the granted project No.02DJ14045)

the volume of emails is much larger and more diverse. With the growth of e-commerce and other electronic services, efficient management of emails for these organizations will be essential.

As a result, some intelligent email classification (EC) methods have been explored to attack this problem, which automatically classify emails into predefined categories and thus greatly facilitate the management of emails. Employing machine learning and information retrieval techniques that learn from email content to classify email flow automatically is desirable with respect to robustness and accuracy. Several methods have been proposed in this field. Among them, there are rule induction approaches [2,3,4], Bayesian approaches [5,6], TFIDF [7,8], decision tree [6]. In experiments of these work, content-based approaches have shown encouraging results.

In more general framework, EC can be viewed as a special case of text classification, but the problem of sparse and noisy feature space makes it more challenging. Compared to general natural language text, email messages are often of short length, and they are noisier due to many informal words and sentences, such as unusual abbreviation, spelling errors etc. Hence the feature space for EC is always far sparser, which presents a difficulty for accurate and stable classification. Moreover, the training samples for EC are usually not in large batch but in dynamic flow with possibly changing content, so the training collection may not provide enough information for the classification algorithms to learn an accurate classifier. While previous work on EC mainly focused on the performance comparison of different classification algorithms [2,3,4,5,6,7,8], they seldom addressed the above special characteristics for EC. We argue that this is the primary limitation for the further success of current EC approaches. To improve the performance of EC, the special characteristics of EC should be carefully dealt with.

In this paper, we propose a novel feature space enriching (FSE) technique to address the above problems of EC. In the process of FSE, semantic features that are related to the terms in the original feature space, are extracted from the concept hierarchies of two semantic knowledge bases (WordNet and HowNet). Then the final enriching features are selected from the semantic features based on the 'information gain increment' filtering metric, and used to enrich the original sparse feature space. The resulting feature spaces will provide semantic-richer features for classification algorithms to learn improved classifiers. Naive Bayes and support vector machine (SVM) are selected as the classification algorithms, which learn classifiers with more accuracy and greater generalization ability from the enriched feature space. Comprehensive experiments on a real-world bilingual (Chinese-English) enterprise email dataset show that: (1) FSE technique can improve the email classification accuracy, especially for the sparse classes, (2) SVM classifier benefits more from FSE than the naive Bayes classifier, (3) with the support of domain knowledge, FSE technique can be more effective.

The rest of this paper is organized as follows. In section 2, we discuss the details of enriching original feature space with semantic knowledge bases. Section

3 describes the classification algorithms used, i.e. naive Bayes and SVM. Experiment results and analysis are presented in section 4. We discuss some related work in section 5. Section 6 concludes with a summarization of this paper as well as some lines of future work.

2 Feature Space Enriching for Email Classification

2.1 Formal Representation of Email Dataset

Like text classification, the first step for email classification is to transform email content into a formal representation suitable for classification algorithms. Traditionally, bag-of-words representation is employed for this task, where an email is represented as a set of words (or terms) and word position is ignored. Thus, the original feature space for email classification can be represented as a list of terms:

$$\vec{V}_o = \{t_1, t_2, \dots, t_k, \dots, t_n\} \quad (1)$$

where t_k is a term that occurs in the email training set \mathbf{T} , n is the total vocabulary size of \mathbf{T} .

Terms differ in their discrimination ability for classification of emails: some are critical for deciding the categories of emails, while others are just inessential. Therefore a proper term weighting scheme is required to encode this difference. In our work, we use a modified vector space model (VSM) [9] from information retrieval community to represent email content. Each email m_i is represented as a weighed term vector as follows:

$$m_i = \{\omega_{1i}, \omega_{2i}, \dots, \omega_{ki}, \dots, \omega_{ni}\} \quad (2)$$

where n is the total vocabulary size of the email training set \mathbf{T} , ω_{ki} is the weight of term t_k in the email training set and is calculated by $TF*IDF$ weighting scheme:

$$\omega_{ki} = TF_{ki} * IDF_{ki} \quad (3)$$

where TF_{ki} is the frequency of term t_k in email m_i . IDF_{ki} is the inverse class frequency which makes the bias that terms appear in many categories are less useful for classification. It is computed as follows:

$$IDF_{ki} = IDF_{kc_j} = \log\left(\frac{N}{n_k}\right) \quad (4)$$

where N is the number of predefined categories, n_k is the number of categories that term t_k occurs in.

2.2 The Utilities of Semantic Knowledge Bases for Email Classification

The naive VSM representation of emails is purely based on the occurrence of terms and totally neglects the semantic and syntactic properties of written text. However, as explained in section one, the feature space for EC is far sparser than in conventional text classification due to the fact that emails are short in length, with noisy terms, and the training instances are usually too few. Therefore, the feature space based on naive VSM representation can't provide enough information for classification algorithms to learn accurate classifiers, and the generalization ability of these classifiers is also very limited due to the sparse feature space. To improve EC performance, additional resources like semantic knowledge base can be used. We use two semantic knowledge bases (one for Chinese terms and another for English terms) to integrating semantic features into the original VSM representation.

The two semantic knowledge bases used are WordNet and HowNet. WordNet is a large online English lexical database that contains various conceptual relations among words [10]. Its basic structure is a concept hierarchy that is composed of a set of nodes and relations among nodes. Each node is a 'synset' (i.e. synonymous set) corresponding to a particular concept and often consists of a number of terms or phrases. Nodes are correlated by various conceptual relations, including hypernymy, hyponymy, meronym etc. A fragment of the WordNet concept hierarchy is shown in Figure 1, which mainly illustrates the hypernymy/hyponymy relation. Hyponymy means 'is a' or 'is a kind of' relation. Hypernymy is inverse of Hyponymy. For instance, lip stick is a hyponym of *makeup*.

HowNet is a public bilingual (Chinese-English) common-sense knowledge base describing relations among concepts as well as relations among the attributes of concepts [11]. The basic concept structure of HowNet is similar to Wordnet. The concepts in HowNet are also represented by a set of Chinese words, and there are also various relations among concepts including hypernymy/hyponymy, synonymy, attribute-host etc. The construction of HowNet is in a bottom-up fashion, and it takes advantage of the fact that all concepts can be expressed using a combination of one or more Chinese characters. HowNet covers about 65,000 concepts in Chinese. Though there are some differences between HowNet and WordNet, we use them in a similar way based on the concept structure in Figure 1.

Generally, related concepts can be represented by a variety of words, and different authors of written text may employ diverse vocabulary. For instance, in our email dataset, the concept of 'transfer goods to customers' can be expressed by the word 'send', or 'ship', 'transport'. Therefore, integrating these features into the representation model will be helpful to enrich the sparse feature space of EC. Fortunately, free open semantic knowledge bases such as WordNet and HowNet greatly facilitate the process. With them the related semantic features such as the synonymies, hypernymy/hyponymy concept features can be imported

to complement the original sparse feature space. This will result in a semantic-richer feature space that can provide more information for learning algorithms and in turn improve the classification performance. We proposed a FSE technique to carry out this process.

2.3 Enriched Feature Space for Email Classification

With the conceptual structure, the semantic knowledge bases can provide rich semantic features for information retrieval tasks. In previous work, they have demonstrated their usefulness for term classification and text classification [12], which motivates us to employ them for enriching the highly sparse feature space of email classification. The main process of FSE is to expand the original feature space with the related semantic features from WordNet and HowNet. The resulted feature space of FSE is defined as follows:

$$\vec{V}_s = \vec{V}_o + \vec{V}_e \quad (5)$$

where \vec{V}_o is the original feature space defined in formula (1), and \vec{V}_e consists of the enriching features extracted from WordNet and HowNet. \vec{V}_o and \vec{V}_e are added to compose the resulted feature space \vec{V}_s . Based on the new populated feature space, the weight of each term in \vec{V}_s will be computed using the formula (3) and (4).

The core procedure of FSE is the selection of the enriching features, as there may be multiple synsets extracted from WordNet or HowNet for each original term, and only some of the synsets have the common concept with the original term. We propose the 'information gain increment' filtering metric to select the enriching features. The intuitive idea behind this scheme is to select only those extracted terms that will increase the global information for learning algorithms. The selection process consists of three steps as follows:

1. For each term in the emails, the synonym and direct hypernymy concepts (with all associated words) are extracted from WordNet (for English term) and HowNet (for Chinese term). Currently we just used hypernymy/hyponymy and synonymy relations, as they are the most common

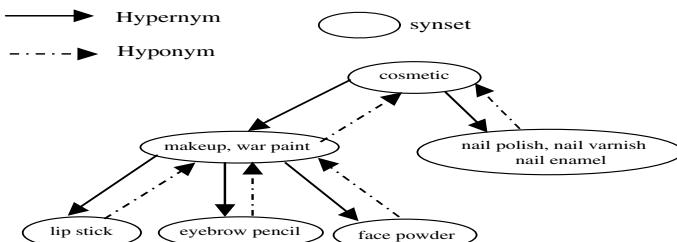


Fig. 1. An Fragment of the WordNet Concept Hierarchy.

relation. During searching, only noun, adjective and verb words are looked up since they are considered semantic-richer.

2. For each extracted term t_k , the information gain of original feature space (denoted as $IG(t_k)_o$) and the enriched feature space (denoted as $IG(t_k)_e$) are calculated by the following formula [13]:

$$IG(t_k) = - \sum_{j=1}^{|C|} P(c_j) \log(P(c_j)) + P(t_k) \sum_{j=1}^{|C|} P(c_j|t_k) \log(P(c_j|t_k)) \\ + P(\bar{t}_k) \sum_{j=1}^{|C|} P(c_j|\bar{t}_k) \log(P(c_j|\bar{t}_k)) \quad (6)$$

where $|C|$ is the total number of the categories, $P(t_k)$ can be calculated from the fraction of emails in which the term t_k occurs and $P(c_j)$ from the fraction of emails that belong to class c_j . $P(c_j|t_k)$ is computed as the fraction of emails from class c_j that term t_k occurs and $P(c_j|\bar{t}_k)$ as the fraction of emails from class c_j that term t_k does not occur.

3. Then the information gain increment of extracted term t_k is computed by:

$$\Delta IG(t_k) = IG(t_k)_e - IG(t_k)_o \quad (7)$$

if $\Delta IG(t_k)$ is positive or higher than some predefined threshold, then the occurrence of extracted term t_k is used to populate the initial feature space.

The result of FSE is a semantic-richer feature space under which the TF*IDF weighting algorithm is performed, and the naive VSM representation is converted into the enriched VSM representation for classification algorithms.

3 Classification Algorithms

We employed Naive Bayes (NB) and SVM as the classification algorithms. The former is a well-known baseline learner, and SVM is one of the best-performed classifiers in text classification.

3.1 Naive Bayes Classifier

NB classifier is a probabilistic classifier based on the statistical independence assumption that features in learning instances are considered conditionally independent given the target label [14]. As for email classification, the task is to label the email m_i with the class c^* which maximizes the conditional probability $P(c^*|m_i)$:

$$c^* = \underset{c_j \in C}{\operatorname{argmax}} P(c_j|m_i) = \underset{c_j \in C}{\operatorname{argmax}} P(c_j)P(m_i|c_j) \quad (8)$$

Directly computing $P(m_i|c_j)$ is difficult since the number of possible vectors for m_i is too high. However, according to the independence assumption of NB that the occurrence of term t_k is independent of the occurrence of any other term t_l , the formula (8) can be replaced by formula (9):

$$c^* = \underset{c_j \in C}{\operatorname{argmax}} P(c_j)P(m_i|c_j) = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{k=1}^{|m_i|} P(t_{ki}|c_j) \quad (9)$$

Learning a NB classifier is just to estimate $P(c_j)$ and $P(t_{ki}|c_j)$ from the training set. $P(c_j)$ can be calculated as the fraction of emails of class c_j in the whole training set, $P(t_{ki}|c_j)$ is calculated from the expanded VSM feature space:

$$P(t_{ki}|c_j) = \frac{1 + \sum_{i=1}^{|c_j|} W_{ki}}{|V| + \sum_{i=1}^{|c_j|} W_i} \quad (10)$$

where $|V|$ is the vocabulary size and severe for Laplace smoothing of the non-occurring terms, W_{ki} is the total $TF*IDF$ weight of term t_k that occurs in the email m_i of class c_j , and W_i is the total weight of all term occurrences in the email m_i of class c_j .

3.2 DDAG-SVM Classifier

SVM was first introduced by Vapnik to solve binary classification problems [15]. Based on the structural risk minimization principle, SVM tries to find the decision hyper-plane between the positive and negative classes with the maximal margin (decided by 'support vectors'). Since a single SVM classifier can only solve two-class classification problems, multiple SVM classifiers must be built for multi-class classification. The traditional methods include one-against-the-rest method that builds N SVM classifiers by label i th class with the positive and all the other example with the negative, or the one-against-one method that constructs all possible two-class classifiers from the N classes. Both of them have no bound on the generalization error and tend to overfit in some cases. For N -class email classification we use the Decision Directed Acyclic Graph (DDAG) algorithm [16] to build an acyclic graph with $N(N - 1)/2$ nodes, each of which is corresponding to a SVM classifiers. DDAG-SVM is very efficient and amenable to a VC-style bound of generalization error, as shown in [16].

4 Experiments and Analysis

In this paper, we carried out experiments on a dataset from a large online cosmetic company. Every day the customer service center receives lots of emails from

Table 1. Email Instance Distribution of The Dataset

Category	Email Number	Percentage
Register	76	4.792
Product	361	22.762
Order	553	34868
Delivery	272	17.150
Payment	235	14.817
Feedback	89	5.611

their customers who write emails in Chinese, English or both. The content of these emails varies from consultation for register problems to inquiry of product information, payment issues etc. To accelerate the service response, the company needs to automatically classify the emails and route them to corresponding personnel. The dataset contains 1586 emails collected from the company mail server and classified into six different categories. Table 1 shows the instance distribution of the dataset.

We developed a scalable email classification system *iMailClass*, which was implemented purely in Java. Accuracy was the main evaluation metric for the experiments.

4.1 Batch-Mode Experiment Results

First, we ran a four-fold cross validation (CV) experiment on the whole dataset. Overall, the classification accuracy of both NB and SVM could be enhanced by our FSE method. The SVM classifier had an increment of 7.58 percent and was more than that of NB classifier which only obtained 5.31 percent increment. We consider that this is due to the fact that SVM has more global optimization and generalization ability so that it can integrate the new features more smoothly and effectively.

Figure 2 shows the detailed results of all categories. We can find that not all categories behave the same with FSE. Categories with few instances, such as 'Register' and 'Feedback', obtain more benefits from FSE. This is due to the fact that their original feature spaces are sparser and the FSE method can supply them more semantic features so that they can provide more discrimination information for learning algorithms. Categories with more emails seem to gain less improvement, such as the category 'Product' and 'Order'. To explain this result, we analyzed the classification confusion matrixes, and found that after FSE more instances of the category 'Product' were misclassified into the category 'Order'. Further inspection on their emails' content revealed that they had quite a few terms in common and the FSE would expand the intersected term set too. In the end, the two categories might be made more confused. Since category 'Order' is more dominant, some similar instances of 'Product' would be classified as 'Order'. In section 4.3 we use domain knowledge to solve this problem.

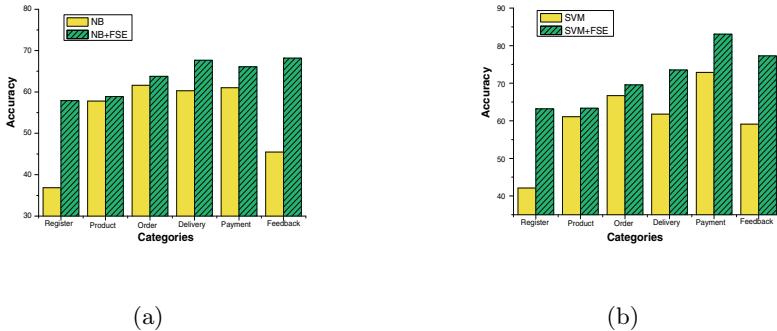


Fig. 2. Detailed Category Accuracy of the CV Experiment: figure (a) is the results of NB classifier, and figure (b) is the results of SVM classifier

4.2 Dynamic Experiment Results

Email domain is dynamic. In many cases training instances for email classification are not in large batch but in dynamic flow with possibly changing content, so the classification system should be able to learn incrementally. To address this problem, we carried out the dynamic experiments. At the beginning, according to the date of email the first 100 ones in the dataset were selected as the initial training instances, then in every step the next 100 emails were taken out to be classified, the classification result of each step was recorded, and these emails were added to training set. This step repeated until the whole dataset had been tested. Figure 3 shows the experiment results on NB classifier and SVM classifier.

It is clear that our FSE method can enhance the classification accuracy of NB as well as SVM. Overall, the accuracy can be increased more during the early stage when the training instances are not enough and the FSE can complement it by integrating features from semantic knowledge bases. With the increasing of training instances, the improvement seems to slow down. As a whole, SVM classifier gains more benefits from FSE than the NB classifier that is similar to the CV experiment. Though the improvement of FSE varies during the dynamic process, FSE is helpful to EC persistently.

4.3 The Effect of Domain Knowledge

Previous work has shown that domain knowledge (or background knowledge) is useful for enhancing the performance of text classification [17,18]. In our experiments, we found that some domain-specific terms with high frequency, such as proper nouns, could not be found in WordNet and HowNet due to the fact that these are general semantic knowledge bases and don't include the words

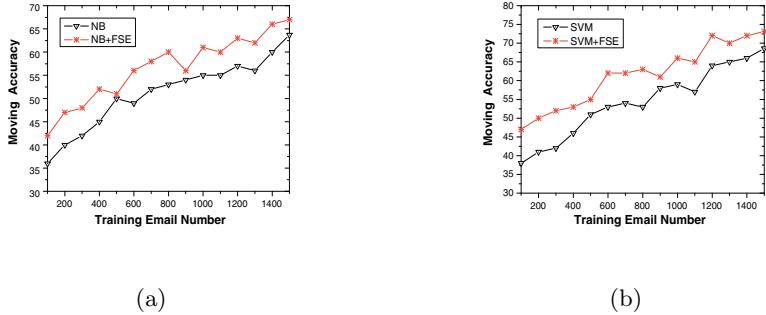


Fig. 3. Dynamic Experiment Results of NB Classifier and SVM Classifier

for special domains. To test if our FSE technique could benefit from the domain knowledge, we extended the WordNet and HowNet with domain knowledge and carried out some preliminary experiments. We expected that extending WordNet and HowNet with domain knowledge would further improve the EC performance.

Currently we just collected domain-specific terms manually from the email samples and some Web pages from cosmetic websites, and grouped them into some synsets based on their co-occurrence frequency in the sample documents. Some synsets like: {‘Chanel’, ‘Christian Dior’, ‘Elizabeth Arden’, … } which is a synset for brand names, and {‘Lumiere’, ‘Hydrabase’, ‘Hydrasoleil’, … } which is a synset for product types. These synsets make up of the domain extension for general semantic knowledge bases, and were used in the FSE process to populate the domain features in the original feature space. Then we ran a four-fold CV experiment on the dataset, and compared the results with that of section 4.1 (without domain knowledge), the experiment results are illustrated in Figure 4.

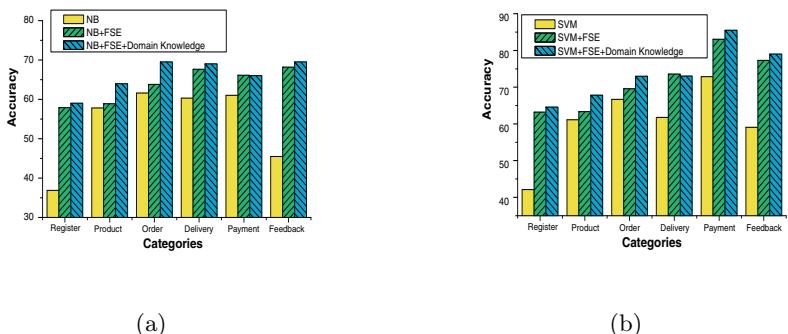


Fig. 4. Experiment Results of the Effect of Domain Knowledge

Overall, the domain knowledge is helpful for improving the classification accuracy. Especially, the category 'Product' and 'Order' benefited much more from it. As stated in section 4.1, the two categories are more confused, which presents a difficulty for FSE. However, with the support of domain knowledge the FSE technique could expand the domain features in the original feature space and increase their term weights, which in turn reinforced their discrimination ability. This implies that our FSE technique can be more effective for enhancing the classification accuracy with the extension of domain knowledge.

5 Related Work

In [2], Cohen used the rule learning algorithm RIPPER to induce sets of keyword-spotting rules that compose the classifiers for classifying email. In the experiments, he compared his method to TFIDF method and got similar accuracy. [4] by Provost also employed RIPPER algorithm for email classification and compared it to Naive Bayes(NB) classifier. All the experiments showed that NB outperformed RIPPER in accuracy. In [3], an ILP framework was used to enable composite rule learner that combined explicit hypothesis construction approach with instance based learning approach. They found the composite rule classifier had the best precision on the items that were classifiable by it.

TFIDF in IR community is another approach that has been extensively used. MailCat [7] employed a TFIDF approach which computed weighted vectors for each class based on word frequencies, then the weighted vectors were used to classify each new message based on the cosine distance. Its improved version SwiftFile[8] addressed the importance of incremental learning for email classification and the experiments showed that incremental learning gained higher accuracy than batch learning. Naive Bayes is another frequently used approach for email classification and anti-spam, and often acts as baseline classifier for evaluating other approaches. The iFile [5] by Rennie is a filter for EXMH mail client based on naive Bayes classification. The experiments on a number of iFile users have shown high performance. Other methods used include decision tree by Diao et al [6], and SVM by Brutlag et al [17].

Most of previous work emphasized on comparing performance of different classification algorithms, and seldom address the special characteristics of email domain as mentioned in section one. Also they mainly focused on personal email classification and seldom addressed the email classification for large enterprises or organizations. These issues were explored in this paper.

6 Conclusion

While previous work seldom addresses the problem of the sparse and noisy feature space for EC, this paper proposes a feature space enriching (FSE) technique to solve this problem. Two semantic knowledge bases, WordNet and HowNet,

are employed to enable the enriching process. The enriched feature space provides semantic-richer feature space for our naive Bayes and SVM algorithms to learn improved classifiers. Experiments on bilingual enterprise email dataset show that the classification accuracy can be improved through this technique, while classes with relative few instances gain more remarkable improvement. We also find that FSE can further improve the classification accuracy with the support of domain knowledge. These experiments demonstrate the effectiveness of our FSE technique for improving EC performance.

There are some lines of work that can be extended in the future. One of them is to test FSE on other classification algorithms such as KNN, neural network etc. Secondly, the FSE technique can also be applied to other text classification tasks which have the similar problem of sparse feature space, such as short message classification, news filtering.

References

1. Whittaker, S., Sidner, C.: Email overload: exploring personal information management of email. In: Proc. of the International conference on human factors in Computing Systems. (1996)
2. Cohen, W.: Learning rules that classify e-mail. In: Proc. of the 1996 AAAI Spring Symposium on Machine Learning and Information Access. (1996)
3. Crawford, E., McCreath, E.: Iems: the intelligent email sorter. In: Proc. of the 19th International Conference on Machine Learning. (2002)
4. Provost, J.: Naive-bayes vs. rule-learning in classification of email. The University of Texas at Austin, Artificial Intelligence Lab, Technical Report (1999)
5. Rennie, J.D.: ifile: An application of machine learning to e-mail filtering. In: Proc. of KDD-2000 Text Mining Workshop. (2000)
6. Diao, Y., Lu, H., Wu, D.: A comparative study of classification-based personal e-mail filtering. In: Proc. of the PAKDD 2000. (2000)
7. Segal, R., Kephart, J.O.: Mailcat: An intelligent assistant for organizing e-mail. In: Proc. of the Third International Conference on Autonomous Agents. (1999)
8. Segal, R., Kephart, J.O.: Incremental learning in swiftfile. In: Proc. of the 17th International Conference on Machine Learning. (2000)
9. Baeza-Yates, R., Ribeiro-Neto: Modern Information Retrieval. ACM Press Series/Addison Wesley,, New York (1999)
10. Miller, G.: Wordnet: a lexical database for english. Communication of ACM **38** (1995) 39–41
11. Dong, Z.: Bigger context and better understanding - expectation on future mt technology. In: Proc. of International Conference on Machine Translation and Computer Language Information Processing. (1999)
12. Gelbukh, A., Sidorov, G., Guzman-Arenas, A.: Use of a weighted topic hierarchy for document classification. In: Lecture Notes in Artificial Intelligence, No.1692: pp. 130-135, Springer-Verlag. (1999)
13. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proc. of the 14th Intl. Conference on Machine Learning. (1997)
14. Mitchell, T.: Machine Learning. McGraw-Hill Series in Computer Science, the United States (1997)

15. Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons Inc. (1998)
16. Platt, J., Cristianini, N., Shawe, J.: Large margin dags for multiclass classification. In: Advances in Neural Information Processing Systems. (2000)
17. Brutlag, J., Meek, C.: Challenges of the email domain for text classification. In: Proc. of the 17th International Conference on Machine Learning. (2000)
18. Zelikovitz, S., Hirsh, H.: Improving short text classification using unlabeled background knowledge. In: Proc. of the 17th International Conference on Machine Learning. (2000)

Synonymous Paraphrasing Using WordNet and Internet*

Igor A. Bolshakov¹ and Alexander Gelbukh^{1,2}

¹Center for Computing Research, National Polytechnic Institute, 07738, Mexico

{igor, gelbukh}@cic.ipn.mx; www.gelbukh.com

²Department of Computer Science and Engineering,

Chung-Ang University, Seoul, 156-756, Korea

Abstract. We propose a method of synonymous paraphrasing of a text based on WordNet synonymy data and Internet statistics of stable word combinations (collocations). Given a text, we look for words or expressions in it for which WordNet provides synonyms, and substitute them with such synonyms only if the latter form valid collocations with the surrounding words according to the statistics gathered from Internet. We present two important applications of such synonymous paraphrasing: (1) style-checking and correction: automatic evaluation and computer-aided improvement of writing style with regard to various aspects (increasing *vs.* decreasing synonymous variation, conformistic *vs.* individualistic selection of synonyms, etc.) and (2) steganography: hiding of additional information in the text by special selection of synonyms. A basic interactive algorithm of style improvement is outlined and an example of its application to editing of newswire text fragment in English is traced. Algorithms of style evaluation and information hiding are also proposed.

1 Introduction

Synonymous paraphrasing (SP) is such change of natural language (NL) text or of its fragments that preserves the meaning of the text as a whole. Nearly every plain text admits SP (in contrast to lists of names, numerical data, poetry, and the like). Computational linguistics has always considered SP an important and difficult problem. The ability of a system to generate good synonymous variations was even considered an indicator of “comprehension” of natural language by computer. Currently, most important applications of SP are text generation and computer-aided style improvement.

There exists a well developed linguistic theory—Meaning–Text Theory by I. Mel’uk [9]—that takes SP as one of its basic principles, considering NL as something like a calculus of synonymous paraphrasings. A set of meaning-conserving rules for restructuring of sentences was developed in frame of MTT. In the process of paraphrasing both words and word order significantly change. The changes in words can

* Work done under partial support of Mexican Government (CONACyT, SNI, CGPI-IPN) and Korean Government (KIPA research professorship). The second author is currently on Sabbatical leave at Chung-Ang University.

touch upon their part of speech and number, for example, *to help tremendously* vs. *to give tremendous help*. However, existing so far special dictionaries and software for paraphrasing based on MTT [1] cover a rather limited fragment of natural language.

Without full-fledged realization of a comprehensive theory of paraphrasing, we nevertheless already possess large linguistic resources—WordNet [8] (EuroWordNet [12]) and Internet—that can help resolving the problem of local paraphrasing with acceptable performance. By local paraphrasing we mean those SP techniques that conserve the structure and word order of a sentence, as well as the number of words (counting stable multiword expressions—or multiwords—like *hot dog* as one unit).

SP is especially important for English—lingua franca of modern sci-tech world. Fortunately, just for English the mentioned resources are highly developed.

In this paper we propose a method of local SP of NL texts based on WordNet synonymy information (synsets) and Internet-based statistics on stable word combinations (collocations) the members of a synset are in. To paraphrase a text, we look for words or multiwords in it that are members of a WordNet synset, and substitute them with other members of the same synset only if they are feasible components of collocations with the surrounding words according to statistical evaluation through the an Internet search engine, such as Google.

The rest of the paper is organized as follows:

- We touch upon the notion of synonymy in order to make it clear that we consider as synonyms not only separate words but also multiwords and that we divide all synonyms into absolute and non-absolute ones, which are used in a different manner for the purposes of SP;
- We describe relevant features of collocations and explain how Internet statistics can be used to test whether two given small text fragments can form a collocation;
- We enumerate and to formalize various types of SP. Some text authors always use only one, the most frequent, synonym for the given concept—unintentionally or to be intelligible for foreigners, children, etc. Other authors widely use synonymous variation to increase literary merits of their texts. So, at least two options are possible for such variation: conformistic (like others) or individualistic. The use of various abbreviations can also be considered a means for SP: some authors and editors prefer concise style, whereas others prefer more verbose one;
- We outline an algorithm realizing interactive SP;
- We present the results of interactive SP applied to a fragment of a newswire text;
- We describe another application of SP methods: given a text, the algorithm analyzes the author's usage of synonyms in it;
- Finally, we develop yet another, totally new application of SP: linguistic steganography, i.e. hiding arbitrarily information in a text by an appropriate choice of synonyms. Namely, each word having synonyms can be replaced by its synonym, depending on the current bit in the bit sequence to be hidden in the text.

2 Absolute and Non-absolute Synonyms

In a simplest definition, synonyms are words that can replace each other in some class of contexts with insignificant change of the whole text's meaning. The references to “some class” and to “insignificant change” make this definition rather vague, but we

are not aware of any significantly stricter definition. Hence the creation of synonymy dictionaries, which are known to be quite large, is rather a matter of art and insight.

A synonymy dictionary consists of groups of words considered synonymous. However, a word can be similar to members of one group in some semantic elements and of another group in other semantic elements. Hence generally speaking a word can belong to more than one synonymy group (if any).

It proved to be insufficient to include to a synonymy groups only separate words: sometimes multiword expressions referring to a given concepts are included. Attempts to translate any dictionary from one language to another always results in the use of such multiwords. For example, the English synonymy group *{rollercoaster, big dipper, Russian mountains}* contain only one single word. Thus we consider multiwords as possible members of synonymy groups.

The only mathematically formal type of linguistic synonymy is when the compared words can replace each other in any context without any change in meaning. These are absolute synonyms, e.g., English *{sofa, settee}*. Absolute synonyms can be formalized as connected by the mathematical relation of equivalence. In the dictionary, such absolute synonyms should be specially marked within their synonymy group.

Note that absolute synonyms are extremely rare in any language. However, there exists much more numerous type of linguistic equivalence—equivalence between various abbreviations and the complete expression. E.g., we can consider as a group of equivalence *{United States of America, United States, USA, US}*. Such equivalents can occur in the same text without any violation of its style. In fact, admission of multiword synonyms brings in a lot of new absolute synonyms like *{former president, ex-president}* or *{comical actor, comic}*.

In many synonymy dictionaries, in each group one member is selected that expresses the common meaning of the words in the group in the most general and neutral way. This, however, is not the case with WordNet [8] and its follower EuroWordNet [12], where all synset members are considered equivalent and corresponding to the common interpretation formula (gloss).

If a word (letter string) enters several synsets, it is always considered homonymous in WordNet, individual homonyms being labeled by different numbers (sense numbers). Homonyms exist in all dictionaries, but in WordNet their quantity is, according to many opinions, exaggerated. In fact, not admitting the same word to enter different synsets does not make all members of each synset absolute synonyms, since there is no guaranty that all of them form the same collocations. Hence, in contrast to absolute synonyms, collocational compatibility with the context should be tested for each member of a synset individually.

As to words commonly recognized as clear homonyms (like *bank*₁ ‘shore’ vs. *bank*₂ ‘organization’), they very rarely enter into the same collocations. So in the task of collocational compatibility of synonyms for a given word, not only all members of its synset but also all members of synsets of its homonyms should be always tested.

Hereafter we assume that a set of synonymy tools is available that includes:

- Synonymy dictionary such as WordNet (or EuroWordNet);
- A specially compiled dictionary of absolute synonyms that contain all above-mentioned types of English equivalents. The synsets of such a dictionary can be subsets of corresponding WordNet synsets, which does not cause any problem since our algorithms look up first absolute synonyms.

So far, WordNet contains rather small number of multiwords, but this number grows from version to version. The discussion in [3] shows that the problem of multiword synonym gathering is fully recognized.

3 Collocations

By a *collocation* we mean a syntactically connected and semantically compatible pair of content words, e.g. full-length dress, well expressed, to briefly expose, to pick up the knife or to listen to the radio; the components of collocations are underlined.

Syntactical connectedness is understood as in dependency grammars [9] and it is in no way merely a co-occurrence of the collocatives in a short span of a text [11]. The head component syntactically governs the dependent component, being adjoined to it directly or through an auxiliary word (usually a preposition). In the linear order of words in the sentence the collocatives can be at any distance from each other, though they are close to each other in the dependency tree.

For a long time collocations were studied in lexicography rather than in computational linguistics. Till now collocations are often treated as series of two or more words occurring together in a narrow window moving along a text [11] or in a specific unit of text [13].

At the same time WordNet includes only semantic links of the paradigmatic type. Their related terms usually include semantically associated components but do not co-occur in close contexts. However, lexicographers have always considered collocations as semantic connections of syntagmatic type with the components usually co-occurring in texts. A comprehensive part of English collocations is now collected in the Oxford Collocations dictionary [10].

To our knowledge, publicly available electronic databases of English collocations did not exist until 1997, when the Advanced Reader's Collocation Searcher (ARCS) for English appeared [2]; however, its deficiencies are too severe for indulgent criticism. The only project in the last decade of a very large collocation database was dedicated to Russian [4]. Thus there is no collocation database for English so far, and though collocation testing could be more easily and reliably done with collocation databases, we have to look for other resources. Just this resource is Internet.

4 Evaluations of Collocation Statistics Via Internet

Hence, our goal is to elaborate a mechanism for assessing whether a word can be replaced with its synonym while keeping collocational cohesion of the text, i.e., a means for deciding which synonyms of a given word can form good collocations with a word in the context.

Consider an example. Suppose the modifying adjective *large-scale* and the noun *project* somewhere to the right of it are found in the same sentence. According to the synonymous dictionary, *large-scale* enters into the synset {*colossal*, *gigantic*, *grandiose*, *great*, *huge*, *large-scale*, *tremendous*, *very large*}. It is necessary to collect statistics in Google on potential collocations that each synonym of the synset could form with the noun *project*.

Google permits collecting statistics only on the number of pages where the two words (or multiwords) co-occur. Only two options of their mutual disposition are measurable: juxtaposition (can be obtained by querying the tested pair in juxtaposition within quotation marks) and arbitrary co-occurrences within a page (queried without quotation marks). The corresponding statistics are given in Table 1.

Table 1. Google statistics of collocations with *project*

Collocation	In quot.	Portion	W/o quot.	Portion	MGV	Portion
<i>colossal project</i>	793	0.5%	123,000	0.5%	9,876	0.5%
<i>gigantic project</i>	2,670	1.7%	255,000	1.0%	26,093	1.3%
<i>grandiose project</i>	1,540	1.0%	83,200	0.3%	11,319	0.5%
<i>great project</i>	80,300	51.6%	9,710,000	38.9%	883,013	44.8%
<i>huge project</i>	34,400	22.1%	4,100,000	16.4%	375,552	19.0%
<i>large-scale project</i>	28,700	18.4%	2,660,000	10.7%	276,300	14.0%
<i>tremendous project</i>	1,620	1.0%	1,340,000	5.4%	46,591	2.3%
<i>very large project</i>	5,570	3.6%	6,690,000	26.8%	193,037	9.8%
Total:	155,593	100.0%	24,961,200	100.0%		

MGV (Mean Geometric Value) in Table 1 is the square radix of the product of numbers obtained in quotation marks and without them. The portion distribution for the collocation set is calculated just for MGVs for the whole set.

As one can see, the ratio between the non-quoted and quoted (sequential co-occurrence, a probable collocation) evaluations is rather big and varies in a broad range. The large ratio values are natural since the non-quoted evaluations count all co-occurrences even at far distance within the page, so that the majority of them do not correspond to collocations of the two components at hand. On the contrary, quoted evaluation corresponds to sequential co-occurrences which probably correspond to collocations. However, not all collocations are counted in this way, since pages with distanced collocations like *great industrial (commercial, political, web, ...)* *project* are not taken into account. Thus the correct number of pages with a given collocation is between the two figures and cannot be measured exactly in this way.

Since only comparative estimations are necessary for our purposes, we evaluate the usage proportions of synonyms within the synset (summing up to 100%) separately for quoted and non-quoted measurements and then take the mean value in each pair of such evaluations. These values are given in the right-hand column of Table 1.

The synonyms with cumulative portion less than a certain threshold μ of marginality are considered unusual in the given collocation and thus not recommended for use in SP. If we take the threshold $\mu = 3\%$, the recommended subset of synonyms in context of *project* is $\{\text{gigantic, great, huge, large-scale, very large}\}$. Just these are the words that we will consider further for various types of the paraphrasing.

Consider now an example where statistics is gathered for a synset's members participating each one in two different collocations: the phrase *heads of various departments*, the synset to be tested being $\{\text{departments, offices, services}\}$; see Table 2.

Table 2. Google statistics of collocations with synonyms of *departments*

Collocation	In quot.	W/o quot.	MGV	Portion
<i>heads of departments</i>	72,700	989,000	268,142	50%
<i>heads of offices</i>	2,320	1,060,000	49,590	12%
<i>heads of services</i>	2,130	5,030,000	103,508	38%
<i>various departments</i>	287,000	5,440,000	1,249,512	34%
<i>various offices</i>	59,000	5,150,000	551,226	17%
<i>various services</i>	297,000	11,200,000	1,823,842	49%

The portion distributions for various collocation sets (in our example, the first three vs. the last three rows in Table 2), are again calculated through MGVs. To combine the data of various sets, we use the mean arithmetic values of the corresponding portions in the different sets. This gives the following distribution:

<i>departments</i>	42%
<i>offices</i>	15%
<i>services</i>	43%

This shows a low portion of *offices*, so this synonym is much less recommendable in this context than the two others (cf. the data of separate collocations). By this composition of tests considered independently, the portion of some synonyms can fall below the marginality threshold.

5 Various Types of Paraphrasing

Paraphrasing can have various objectives. Having in mind the first example in the previous section as illustration, we can classify its types as follows.

Text compression. For this, the shortest synonym is taken in each synset (either independently of any statistical evaluations or selecting from the words that passed the marginality threshold). In our example, this is *huge*. This gives a significant gain in space only when there are abbreviation(s) among absolute synonyms.

Text canonization. For this, the most frequently used synonym is taken. Of course, it may prove to be the same one as in the source text. In our example, this is *great*. The text becomes more canonical—or banal, without variations. It is useful from the viewpoint, say, of legislative bodies, since in the legislation even common words can be considered strict terms. It is also useful for persons with limited language knowledge, i.e. for foreigners or children, since this renders texts in a more intelligible way.

Text simplification. Any text will be more intelligible for language-impaired person if we select among synonyms a “simpler,” colloquial variant [5]. It is not always the most frequently used synonym, though in our example this is probably also *great*. We consider language-impaired persons as native adults with rather low educational level whose language abilities scarcely could be improved.

The algorithm of synonymous paraphrasing for the simplification is roughly as follows. If for a given word there are any synonyms marked in the dictionary as *colloquial*, we select the most frequent one of them. Otherwise, if there are any neutral synonyms (without any stylistic mark), we select the shortest one of them, assuming that the shortest is the simplest for average language-impaired person’s mentality. In

particular, in this way the scientific, bookish or obsolete words will be substituted with colloquial or neutral synonyms.

Conformistic variations. For this, the synonym is taken randomly with the distribution corresponding to the frequencies obtained through Internet evaluations. Such a choice fully corresponds to the present usage of the given synset's members.

Individualistic variations. We may imagine individualistic (counter-conformistic) variation as selection of the most rarely used option among those exceeding the marginality threshold. Since the value of the threshold is taken on rather subjective grounds, this tactics may be considered risky and sometimes gives erroneous results.

6 Basic Algorithm of Interactive Paraphrasing

Below we outline—with significant simplifications, especially as to the conformistic style mode—the interactive SP procedure.

1. Ask $mode \in \{compression, canonization, simplification, conformism, individualism\}$
2. Ask marginality threshold $\mu \in (0,1)$ and sensitivity threshold $\lambda \in (0,1)$
3. For each non-functional word or multiword w which is a member of a synset
4. Let $S = \text{union of all relevant synsets } s \text{ for } w$
5. For each word v in S
 6. If its appropriateness $a(v) < \mu$ then set $score(v) = 0$
 7. Else
 8. If $mode = compression$ then set $score(v) = 1 / length(v)$
 9. If $mode = canonization$ then set $score(v) = a(v)$
 10. If $mode = simplification$ then set $score(v)$ as described in Section 0
 11. If $mode = conformism$ then set $score(v) = random$ from 0 to $a(v)$
 12. If $mode = individualism$ then set $score(v) = 1 / a(v)$
 13. If $score(w) / \max_s score(v) < \lambda$ then
 14. Suggest to the user all variants v in S , $score(v) \neq 0$, in the order of $score(v)$

By relevant synsets in line 4 we refer to a word sense disambiguation procedure if it is available; otherwise all senses are considered relevant. Since we cannot distinguish between (relevant) senses, we have to consider all members of all such synsets to be equally possible synonyms of the given word, hence the union; however, the synonyms of wrong meanings are unlikely to pass the marginality threshold in line 6.

Appropriateness is determined as described in Section 0. If syntactic heuristics selecting possible dependency links for a given word are available, the context words to try the collocations with are selected accordingly. Otherwise, all non-functional words in the same sentence within certain linear distance from the given word are used.

The condition in line 13 is needed to force the algorithm to suggest corrections only where they are really necessary and not at every word.

After the work has been finished, the user can assess the result as described in Section 0 and compare the obtained score with that of the optimal transformation consisting in automatically accepting the variant with the highest score for each word.

7 An Experiment on Text Fragment Paraphrasing

For a real experiment with SP, an English fragment from a Russian newswire site Gazeta.ru was taken. Our initial assumption was that the translators from Russian to English from the Russian news agencies are not as skilled in the language as their native English-speaking colleagues, so that the results of paraphrasing might be of practical interest. The fragment taken was as follows:

The Georgian foreign_minister (foreign_office_head) is scheduled (planned, designed, mapped out, projected, laid on, schemed) to meet (have a meeting, rendezvous) with the heads (chiefs, top_executives) of various (different, diverse) Russian departments (offices, services) and with a deputy of Russian foreign_minister (foreign_office_head). “Issues (problems, questions, items) concerning (pertaining, touching, regarding) the future (coming, prospective) contacts at the higher (high-rank) level will be discussed (considered, debated, parleyed, ventilated, reasoned, negotiated, talked about) in the course of the meeting (receptions, buzz sessions, interviews).” said Georgian ambassador to Russia Zurab Abashidze. The Georgian foreign_minister (foreign_office_head) will be_in (visit) Moscow on a private (privy) visit (trip), the Russian Foreign Ministry reported (communicated, informed, conveyed, announced).

Let us discuss the transformations listed in Section 0 as applied to this text.

Text compression. The potential improvements are: *scheduled* → *planned*, *departments* → *offices*, *issues* → *items*, *concerning* → *touching*, *discussed* → *debated*, *private* → *privy*, *visit* → *trip*. Not all of them would pass the statistical test. For example, a combination *foreign minister is scheduled* is 60 times more frequent than *foreign minister is planned*.

Text canonization. Our tests showed that a few words can be changed in the text: (*will*) *be_in* → *visit (Moscow)*, 3 times more frequent; (*Ministry*) *reported* → *announced*, 1% more frequent. On the other hand, in most cases the translator has chosen the correct synonym. For example, *issues* is 3 times more frequent with *concerning* than *problems*; *future* 20 times more frequent than *prospective* with *contacts*; *visit* 13 times more frequent than *trip* with *visit*. Thus, the overall quality of translation in the text under consideration can be assessed as quite good.

Text simplification. Here, the first candidates to substitution are the words having colloquial variants: *discussed* → *talked about* and *meetings* → *buzz sessions*. The other substitutions are the same as for text compression.

Conformistic variations. Here is a possible variant of such SP:

The Georgian foreign_office_head is planned to have a meeting with the heads of diverse Russian offices and with a deputy of Russian foreign_office_head. “Questions touching the future contacts at the high-rank level will be debated in the course of the interviews,” said Georgian ambassador to Russia Zurab Abashidze. The Georgian foreign_minister will visit Moscow on a private trip, the Russian Foreign Ministry informed.

Individualistic variations. Here is a possible variant of this type of SP:

The Georgian foreign_office_head is projected to rendezvous with the top_executives of diverse Russian departments and with a deputy of Russian foreign_office_head. “Issues

regarding the prospective contacts at the high-rank level will be parleyed in the course of the receptions," said Georgian ambassador to Russia Zurab Abashidze. The Georgian foreign office head will visit Moscow on a privy visit, the Russian Foreign Ministry conveyed.

8 Another Application: Style Evaluation

The most usual way to evaluate the style of the text is currently through easily gathered statistics of word length in letters, sentence length in words, and paragraph length in sentences. This is too formalistic to give good results.

Meantime, the use of synonyms can evidently estimate an important aspect of the literary style. For example, repeated use of the same synonym for the given notion makes the text banal and dull, though maybe good for technical writing. Diverse but conformistic use of synonyms considered by many a good literary style, but some journalists prefer counter-conformism (cf. Section 6).

So we suppose that a user of an automatic style checker wants to obtain the evaluation in points that assess four characteristics: compressibility, variation, conformism, and individualism.

The algorithm for assessing compressibility works (with some simplifications) as follows.

1. Set *Compressibility* to 0
2. For each non-functional word w in the text
3. Set $S = \text{union}$ of all relevant synsets containing w
4. Remove from S the members v with appropriateness $a(v) < \mu$
5. Let v_0 be the shortest word in S
6. Increase *Compressibility* in $\text{length}(w) - \text{length}(v_0)$

Again, by relevant synsets we refer to a word sense disambiguation procedure if it is available; otherwise, all synsets are considered relevant. By appropriateness we refer to the procedure discussed in Section 0, where μ is the marginality threshold.

For measuring variation, conformism, and individualism, we need to compare the usage statistics g in Internet and f in the given text, for each word used in the text.

1. Consider only synsets relevant for at least one non-functional word in the text
2. For all words w in all synsets s
3. Set $g_s(w) = 0$ (Google statistics)
4. Set $f_s(w) = \text{the number of occurrences of } w \text{ for which } s \text{ is relevant}$
5. or $f_s(w) = 1$ if no occurrences (for smoothing)
6. Set $\varphi_s(w) = 1 / f_s(w)$ (inverse frequency, for individualism)
7. For each occurrence of a word w
8. For each synset s relevant or it
9. For each member of s
10. Increase $g_s(w)$ in the frequency obtained from Internet
11. For each synset s
12. Normalize g_s, f_s, φ_s within s so that $\sum_w g_s(w) = \sum_w f_s(w) = \sum_w \varphi_s(w) = 1$
13. Set *variation* to average dispersion of $f_s(w)$ within synsets
14. Set *conformism* to average cosine similarity between g_s and f_s
15. Set *individualism* to average cosine similarity between g_s and φ_s

By Internet statistics in line 10 we again mean the procedure described in Section 0, which depends on the context of each specific occurrence of a word, which we implicitly average across all occurrences.

The above procedure generates the absolute value of the corresponding characteristic. What is more interesting for the user, however, is the relative value: is the text optimal or can be significantly improved? This can be assessed as the ratio between the absolute score obtained for the given text and that of a text optimized as described in Section 0 by automatically choosing the best variant at each text position.

9 Yet Another Application: Linguistic Steganography

Linguistic steganography [6, 7] is a set of methods and techniques permitting to hide within a text any binary information, based on some purely linguistic knowledge. For hiding the very fact of enciphering, the resulting text should not only remain innocuous but also conserve grammatical correctness and semantic cohesion. For hiding information, some words in the source text are replaced by other words in the way controlled by the bit sequence to be hidden and detectable at the receiver's side. In the best case, the resulting text conserves the meaning of the source one.

Chapman *et al.* [7] proposed to take beforehand a synonymy dictionary and enumerate the words within each its group. When their steganographic algorithm encounters in the text a word from one of these groups, it replaces the word by its synonym having intra-group number equal to the binary content of a small current portion of the information to be hidden. It is clear that while scanning the resulting text, the reverse algorithm will find the representatives of just the same synonymy groups and restore the hidden information basing on their numbers within the groups.

The described idea does work, but context-independent synonymous changes usually do not preserve the meaning. Additionally, the resulting texts become semantically non-cohesive (incomprehensible) and thus can loose their initial innocuity.

We propose to divide the synonyms into two large classes. Synsets of absolute synonyms are used in the same context independent manner. However, the synsets of non-absolute synonyms are previously tested for conforming to collocations in the text the source synonym is in. Only those non-absolute synonyms that pass all collocational tests form the relevant groups are used. The inner numbers within these (usually reduced) groups are taken for steganography.

The proposed steganographic algorithm has two inputs:

- The source natural language text of the minimal length of approximately 200 per bit of the information to be hidden. The text format can be arbitrary, but it should be orthographically correct, to avoid later corrections by someone else. The text should also be semantically “common,” since the presence of lists of names, sequences of numbers, and the like increase the text length required for hiding.
- The information to hide, merely as a bit sequence.

The steps of the algorithm are as follows:

Search of synonyms. From left to right, (multi)words that are entries of the synonymy dictionary are extracted (in case of ambiguity, the longest multiword is taken).

Formation of synonymy groups. The synsets are analyzed one by one. If the synset consists of absolute synonyms, only they are immediately taken and ordered in a predetermined manner (e.g., alphabetically). If this is a synset of non-absolute synonyms, then it is checked whether the textual synonym is homonymous and its homonyms are the members of other synsets. All newly found homonyms are grouped for further collocation proving.

Collocation proving of synonyms. All members of non-absolute synsets are checked against their context, group by group. The context words in the text that could form a collocation with members of the tested synset are sent as a query to Google. Each query is sent in two forms, in quotation marks and without them. The statistics obtained is normalized in the manner described in Section 4. Each pair {synonym, context word} is statistically evaluated against Internet as a pair of components of a collocation. If the synonym has several senses, all of them are tested. If the context word is absolutely synonymous or not synonymous, the tests are carried out only with it. Otherwise (if the context word belongs to a group of non-absolute synonyms), the tests are done with all of them. At each step, the synonym under test is excluded from its group if a certain threshold μ is not reached. The synonyms that pass this test are ordered within the reduced synsets in the predetermined manner. All non-functional context words, both to the left and to the right from the current word, are taken within the current sentence.

Enciphering. The sequence of the obtained synonymy groups is scanned from the left to the right. The current group is cut in length to the nearest power p of 2. The next piece of length p is picked up from the bit sequence to be hidden, and the synonym is taken with the number equal to this piece. It replaces the synonym in the source text. If grammatical features of the newcomer (number or person, depending on the part of the speech) differ from the source word, special operations of agreement are performed.

This process continues until one of the inputs is exhausted. In normal situation, the hided information sequence ends earlier and hereafter the source text does not change.

10 Conclusions and Future Work

We have proposed a method for synonymous paraphrasing of natural language texts by contextually controlled synonymous variation of individual words. We quantitatively characterize the naturalness (appropriateness) of a word or its synonyms in the given context by the frequency of collocations of the given word with the words from the context. This frequency in the general texts is estimated as their relative frequency in Internet, using an Internet search engine. As a synonymy dictionary we use WordNet.

We have pointed out at least two practical applications of our method. The first one is style checking and correction. For each word in the text, we generate its possible synonymous variants appropriate in the given context; if there are much better variants, we suggest them to the user as possible improvements. What is more, comparing the average appropriateness of words in the given text with that of the best word choice at each position generated automatically, we can assess the stylistic quality of the given text as optimal or allowing significant improvement.

The second application is linguistic steganography: hiding arbitrary information within a text without changing its grammaticality, cohesion, and even meaning. Recently this has been an active research area having in its turn a number of important applications. One of them is a way of secret communication in the situation when the very fact of communication is to be kept secret (and thus, usual cryptographic methods prove insufficient). Another one is watermarking: digitally “signing” the text in such a way that the signature can be restored even after significant and probably intentional transformations of the text or its format (e.g., plagiarism).

In the future, we plan to consider other applications of the suggested ideas, e.g., word choice in automatic translation. Also, the measure of the linguistic appropriateness is to be extended to better take into account various linguistic phenomena.

References

1. Apresian, Ju. D., et al. *ETAP-3 Linguistic Processor: a Full-Fledged NPL Implementation of the Meaning–Text Theory*. Proc. First Intern. Conf. Meaning–Text Theory, MTT 2003, Paris, Ecole Normale Supérieure, June 2003, p. 279–288.
2. Bogatz, H. *The Advanced Reader’s Collocation Searcher (ARCS)*. ISBN 097093414-9, www.asksam.com/web/bogatz, 1997.
3. Bentivogli, L., E. Pianta. *Detecting Hidden Multiwords in Bilingual Dictionaries*. Proc. 10th EURALEX Intern. Congress, Copenhagen, Denmark, August 2002, p. 14–17.
4. Bolshakov, I. A., *Getting One’s First Million... Collocations*. In: A. Gelbukh (Ed.) *Computational Linguistics and Intelligent Text Processing*. Proc. 5th Intern. Conf. on Computational Linguistics CICLing-2004, Seoul, Korea, February 2004. Lecture Notes in Computer Science No. 2945, Springer, 2004, p. 229–242.
5. Carroll, J., G. Minnen, D. Pearse, Y. Canning, S. Delvin, J. Tait. *Simplifying text for language-impaired readers*. Proc. 9th Conference of the European Chapter of the ACL EACL’99, Bergen; Norway, June 1999.
6. Chapman, M., G. Davida. *Hiding the hidden: A software system for concealing ciphertext as innocuous text*. In: Yongfei Han, Tatsuaki Okamoto, Sihan Qing (Eds.) Proc. 1st Intern. Conf. on Information and Communication Security ICICS 97. Lecture Notes in Computer Science 1334, Springer, 1997, p. 335–345.
7. Chapman, M., G. I. Davida, M. Rennhard. *A Practical and Effective Approach to Large-Scale Automated Linguistic Steganography*. In: G. I. Davida, Y. Frankel (Eds.) *Information security*. Proc. of Intern. on Conf. Information and Communication Security ICS 2001, Lecture Notes in Computer Science 2200, Springer, 2001, p. 156–165.
8. Fellbaum, Ch. (Ed.) *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
9. Mel’uk, I. *Dependency Syntax: Theory and Practice*. SONY Press, NY, 1988.
10. *Oxford Collocations Dictionary for Students of English*. Oxford University Press. 2003.
11. Smadja, F. *Retrieving Collocations from text: Xtract*. Computational Linguistics. Vol. 19, No. 1, 1990, p. 143–177.
12. Vossen, P. (Ed.). *EuroWordNet General Document*. Vers. 3 final. www.hum.uva.nl/~ewm.
13. Biemann, C., S. Bordag, G. Heyer, U. Quasthoff, C. Wolff. *Language-independent Methods for Compiling Monolingual Lexical Data*. In A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing (CICLing-2004)*. Lecture Notes in Computer Science, N 2945, Springer, 2004, pp. 214–225.

Automatic Report Generation from Ontologies: The MIAKT Approach

Kalina Bontcheva and Yorick Wilks

Department of Computer Science, University of Sheffield
211 Portobello St, Sheffield, UK S1 4DP
{kalina,yorick}@dcs.shef.ac.uk

Abstract. This paper presented an approach for automatic generation of reports from domain ontologies encoded in Semantic Web standards like OWL. The paper identifies the challenges that need to be addressed when generating text from RDF and OWL and demonstrates how the ontology is used during the different stages of the generation process. The main contribution is in showing how NLG tools that take Semantic Web ontologies as their input can be designed to minimises the portability effort, while offering better output than template-based ontology verbalisers.

1 Introduction

The Semantic Web aims to add a machine tractable, re-purposeable layer to complement the existing web of natural language hypertext. In order to realise this vision, the creation of semantic annotation, the linking of web pages to ontologies, and the creation, evolution and interrelation of ontologies must become automatic or semi-automatic processes.

Natural Language Generation (NLG) takes structured data in a knowledge base as input and produces natural language text (see [1]). In the context of Semantic Web or knowledge management, NLG can be applied to provide automated documentation of ontologies and knowledge bases or to generate textual reports from the formal knowledge. Unlike human-written texts, an automatic approach will constantly keep the text up-to-date which is vitally important in the Semantic Web context where knowledge is dynamic and is updated frequently. The NLG approach also allows generation in multiple languages without the need for human or automatic translation [2].

The main challenge posed for NLG by the Semantic Web is to provide tools and techniques that are extendable and maintainable (the majority of existing NLG applications can only be modified and extended by NLG experts).

This paper presents the MIAKT (Medical Imaging and Advanced Knowledge Technologies) project where NLG is used to generate automatically reports from knowledge encoded in the domain ontology (Section 2). Section 3 discusses the MIAKT generator, followed by performance evaluation results (Section 4) and a discussion on domain portability (Section 5). Finally, Section 6 presents some related work, and Section 7 outlines directions for future work.

2 MIAKT

This work was carried out as part of the e-science project MIAKT¹, which aims at developing Grid enabled knowledge services for collaborative problem solving in medical informatics. In particular, the domain in focus is Triple Assessment in symptomatic focal breast disease.

The role of NLG in the project is to generate automatically textual descriptions from the semantic information associated with each case - patient information, medical procedures, mammograms, etc. The reports are aimed at the medical professionals involved in the diagnosis and treatment, therefore it is essential to convey in the generated report the complete information available in the ontology about the patient.

The majority of semantic information is encoded in the domain ontology, which is a formal description of the breast cancer domain [3] and is encoded in DAML+OIL [4]. In addition, each case has a case-specific, i.e., instance knowledge, which is encoded in RDF [5] and specifies information about this particular case, e.g., which medical procedures were undertaken, sizes and locations of lesions, diagnosis. The domain ontology was engineered manually as part of the project and does not contain inconsistencies. This is frequently the case with medical ontologies and terminological lexicons as they are then used as standards in the community (e.g., UMLS [6]).

In order to avoid the cost of having to parse and represent ontologies in each of these formats (DAML+OIL and RDF) in MIAKT, we used GATE's ontology tools [7] that can parse these formats and convert them into a common object-oriented model of ontologies with a unified API (Application Programming Interface). Consequently, our generator was developed to work from this common representation, in isolation from the concrete ontology implementation. The benefit of this approach is that if a new ontology format needs to be added at a later date (e.g., OWL), the generator would not need to be modified.

We used the NLG lexicon tools [8] to create a lexicon of 320 terms in the domain of breast cancer and map them to the 76 concepts and 153 instances in the MIAKT ontology. These terms were collected manually from the BIRADS lexicon of mammography terms² and NHS documents³, then verified and enriched manually with synonyms from online papers, Medline abstracts, and the UMLS thesaurus [6].

3 The GATE-Based MIAKT Generator

The MIAKT generator takes as input the medical ontology, an RDF description of the case, and the MIAKT NLG lexicon. The output is a textual report

¹ Project Web site: <http://www.aktors.org/miakt>. MIAKT is supported by the UK Engineering and Physical Sciences Research Council as part of the MIAKT project (grant GR/R85150/01), which involves the University of Southampton, University of Sheffield, the Open University, University of Oxford, and King's College London.

² Available at http://www.acr.org/departments/stand_accred/birads/

³ <http://www.cancerscreening.nhs.uk/breastscreen/index.html>

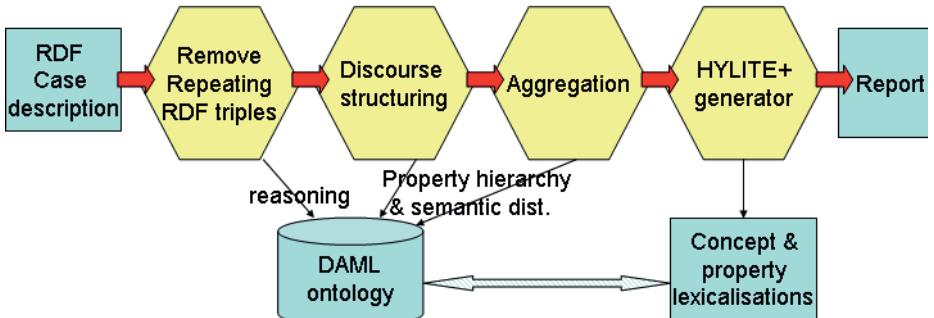


Fig. 1. The MIAKT Generator

verbalising the semantic information provided for the given case, enriched with information from the ontology and using the terms provided in the MIAKT lexicon.

The RDF case description is created by the medical professional who carried out the examination and made the diagnosis. MIAKT provides an ontology-based user interface where the medics can annotate the images (e.g., X-rays, MRI scans) with semantic information (e.g., shape of tumor, size, diagnosis) and also enter patient details, etc.

In order to generate natural language reports from RDF, there are several challenges that need to be addressed:

- The RDF can sometimes contain repetitive information, due to presence of inverse relations in the ontology, e.g., `involved_in.ta(01401_patient, ta-1069861276136)` and `involve_patient(ta-1069861276136, 01401_patient)`, or due to one fact entailing another. Therefore the generator needs to use a reasoner based on the ontology to filter out repetitive RDF statements.
- Need to order the RDF triples in well-structured discourse.
- If each triple is used to generate a separate sentence, the text becomes hard to read, with too many short sentences. Therefore, aggregation needs to be performed as part of the generation process to combine several triples into one sentence.
- Finally, if a sentence is being formed from more than one triple, the surface realiser needs to determine whether to use conjunctive or subordinate clauses; which attributive properties should be expressed as adjectives and which as nouns; when to use passive or active voice, etc.

The MIAKT generation architecture is shown in Fig. 1. The first stage is to remove already verbalised triples from the RDF (Sect. 3.2), then the remaining RDF triples are ordered to form well-structured discourse (Sect. 3.3), followed by an aggregation stage where similar RDF triples are combined to form one sentence, resulting into a more coherent text. Finally the text is produced by

the ontology-based verbaliser, which uses the lexicon and the property hierarchy from the ontology. We will focus on that process next.

3.1 The Ontology-Based Realiser

The ontology-based realiser transforms RDF statements into conceptual graphs (a kind of semantic network) which are then verbalised by the HYLITE+ surface realiser [9]. The output is a textual report.

The surface realiser does not use templates (cf [10] for template-based verbalisation of RDF), i.e., have fixed expressions where the arguments of each relation are inserted. Instead, its input is the RDF statement and the concept which is going to be the subject of the sentence. Then it treats the RDF as a graph and finds a path through that graph, starting from the given concept and visiting all properties and their arguments. For further details see [9].

The HYLITE+ surface realiser already has a list of 40 different relations that it can verbalise. Some are linguistically motivated relations like AGNT (agent), PTNT (patient), and OBJ (object), while others describe attributes, locative relations, part-whole relations, etc. When these relations are compared to the properties in ontologies, there is a substantial gap, because properties are typically not linguistically motivated (e.g., `has_date`, `produce_result`). In order to verbalise such properties the surface realiser needs a lexicalisation for each one of them.

In order to make the ontology-based surface realiser more portable and reduce the need for specifying manually the lexicalisation of each property in the ontology, we defined a core set of 4 basic property types – `active-action`, `passive-action`, `attribute`, and `part-whole`. In the MIAKT ontology any other property is defined as a sub-property of one of these 4 generic ones.

Sub-types of attribute and part-whole properties can then be handled automatically, because they correspond to `PART_OF` and `ATTR` relations and the realiser already has grammar rules to verbalise them. New properties can be introduced in the ontology as sub-properties of either attribute or part-whole property and they will be handled automatically by the surface realiser.

Active action and passive action properties were introduced to help the surface realiser with mapping the arguments of such properties to semantic roles like agent, patient, and object. In general, the first argument of active properties is mapped to an agent role, whereas the first argument of passive properties is mapped to a patient role. For example the triple from case0140 (see Figure 2) involving the active property `produce_result`:

```
<rdf:Description rdf:about=
    'file:/...#01401_mammography'>
<NS2:produce_result rdf:resource=
    'file:/...#image_01401_left_cc'/>
... </rdf:Description>
```

is mapped to two facts using the `AGNT` and `OBJ` relations and a concept `PRODUCE_RESULT`:

```
AGNT(Mammography: 01401_mammography, PRODUCE_RESULT)
OBJ(PRODUCE_RESULT, Medical_Image: image_01401_left_cc)
```

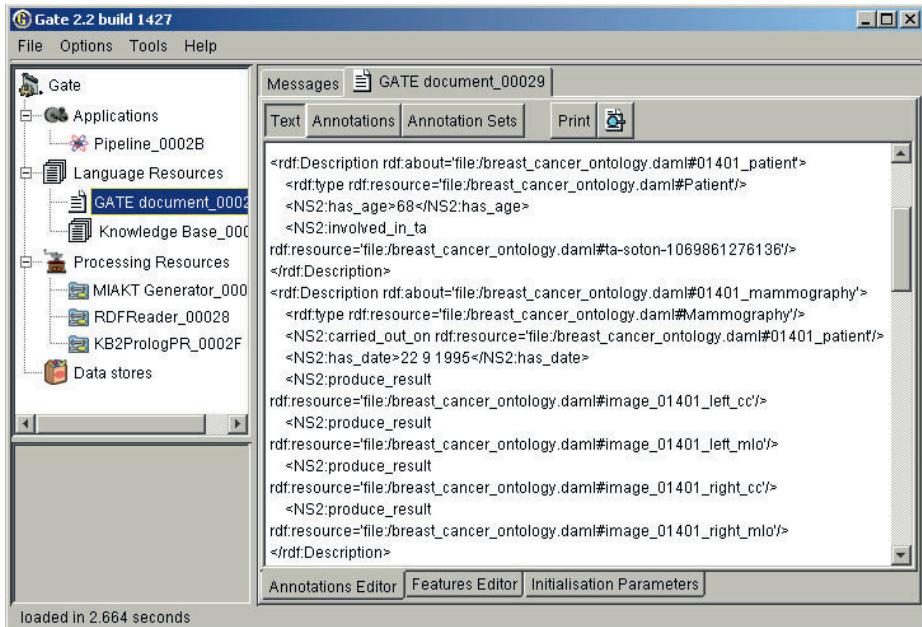


Fig. 2. The case information encoded as RDF (case0140)

The MIAKT lexicon already contains lexicalisations for concepts and instances that occur in the ontology, i.e., mammography and medical image in this case. Therefore, we additionally specified the lexicalisation of each active and passive action property (e.g., PRODUCE_RESULT). At present, there are only 4 active action and 3 passive action sub-properties in the ontology, so it was not a substantial overhead. If new active or passive action properties are added in the future, their lexicalisations need to be specified manually by the ontology builder in order to enable their verbalisation. However this is not a difficult task, as the user only needs to provide the verb that corresponds to the new property in its base form and the generator will deal with the morphological issues.

To summarise, the ontology-based realiser was made more generic and easier to adapt by introducing a property hierarchy in the ontology, based on 4 linguistically motivated basic types. This hierarchy also plays an important role in allowing the creation of more generic text schemas (Section 3.3). The decision to introduce linguistically motivated properties in the ontology can be seen as undesirable in some applications, so it is possible to introduce this classification only in the generator's input handler, leaving the original ontology intact.

Linguistically oriented ontologies have already been used as interface methods between generators and formal knowledge of the domain in some NLG systems (e.g., ONTOGENERATION [2]). The most frequently used one is the Generalised Upper Model (GUM) [11], which is a linguistic ontology with hundreds of concepts and relations, e.g., part-whole, spatio-temporal, cause-effect. In con-

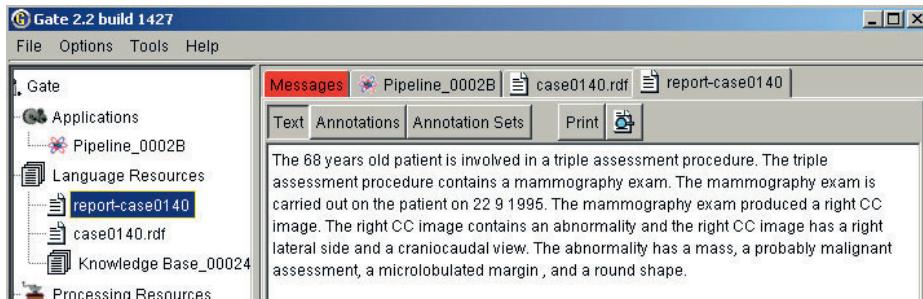


Fig. 3. The generated report for case0140

trast, in MIAKT we chose to use few, very basic distinctions in order to enable knowledge engineers rather than NLG experts to perform the mapping between properties and their linguistic expression. Our experience indeed showed that the ontology engineer found the distinction between the 4 basic types intuitive to understand and use [3]. However, the choice of only few basic types has an impact on the variability and expressivity of the generator. This has not posed a problem in our domain, because the majority of the information is conveyed in the noun phrases and also the reports tend to have a fairly regular structure.

3.2 Filtering Out Repetitions Using the Ontology

The report generation process starts off by being given a set of triples describing a case, in the form of an RDF file (see Figure 2). Since there is some repetition, these triples are first filtered to remove already said facts. In addition to triples that have the same property and arguments, the system also removes triples involving inverse properties with the same arguments, as those of an already verbalised one. The information about inverse properties is provided by the ontology – both DAML+OIL and OWL have this primitive for properties.

For example, the MIAKT ontology states that these two properties are inverse:

```

<daml:ObjectProperty rdf:about=
    "file:/...#involved_in_ta">
  <daml:inverseOf rdf:resource=
    "file:/...#involve_patient"/>
...

```

At the start of the report, the generator verbalises the triple `involved_in_ta(01401_patient, ta-soton-1069)` as part of the first sentence (Figure 3) describing the patient. Then the following triples are provided in the RDF input for the given instance of triple assessment procedure:

```

involve_patient(ta-soton-1069, 01401_patient)

consists_of_subproc(ta-soton-1069, 01401_mammography)

```

Since the ontology specifies that `involve-patient` and `involved_in_ta` are inverse properties and the triple `involved_in_ta(01401_patient, ta-soton-1069)` has already been verbalised, the `involve-patient` triple is removed as already known.

In this way, the ontology plays an important role in detecting repetitive input. Future work will experiment with removing implied information, which logically follows from already said facts. This will be done by using a reasoner and axioms from the ontology, in order to determine which statements follow from other ones. Such functionality is often provided by semantic repositories such as Sesame (<http://www.openrdf.org/>).

3.3 Discourse Planning and the Property Hierarchy

The main effort in porting a generator to a new domain is in the adaptation of its discourse planning component. The example report in Figure 3 shows that typically the MIAKT reports start off by describing the patient, then move on to the medical procedures and their findings. The RDF input for the case however (Figure 2) does not specify any ordering between the RDF triples. In order to bridge this gap, we created discourse patterns that are applied recursively by the discourse structuring algorithm and capitalise on the property hierarchy.

The top-level schema looks like:

```
Describe-Patient ->
  Patient-Attributes,
  Describe-Procedures
```

where `Patient-Attributes` and `Describe-Procedures` are recursive calls to other schemas. For example, the `Patient-Attributes` schema collects recursively all properties that are sub-properties of the `attribute-property` and involve the given patient:

```
Patient-Attributes ->
  [attribute(Patient, Attribute)],
  Patient-Attributes *
```

As can be seen, the discourse schemas exploit the property hierarchy in order to be independent of property names in the given ontology. However, if more specialised treatment of some properties is required, it is possible to enhance the schema library with a new pattern, that applies only to a specific property.

In the example shown in Fig. 2, only one attribute property involving the patient is matched by the `Patient-Attributes` schema: `has_age(01401_patient, 68)`. The `Describe-Procedures` schema then matches the `involved_in_ta(01401_patient, ta-soton-1069)` triple and continues by recursively describing each of the procedures from the matched triples (in this case – one triple assessment procedure).

3.4 Ontology-Based Aggregation

Once the information from the ontology is structured using the schemas, the next stage is to perform aggregation and join similar RDF triples. This aggregation is done at discourse level [12] by joining adjacent triples that have the same first argument and have the same property name or if they are sub-properties of **attribute** or **part-whole** properties. For example, in the last sentence in Figure 3 we have 4 triples with the same first argument (the abnormality) and all properties are attribute properties. Therefore, they are joined together as one proposition, which gives rise to the last sentence. The aggregated proposition is which is then passed to the surface realiser (Sect. 3.1):

```
ATTR(Abnormality: 01401_abnormality, Mass: 01401_mass)
ATTR(Abnormality: 01401_abnormality, Margin: inst_margin_microlob)
ATTR(Abnormality: 01401_abnormality, Shape: inst_shape_round)
ATTR(Abnormality: 01401_abnormality, Diagnose: inst_ass_prob_malig)
```

Without this aggregation step, there will be four separate sentences, resulting in a less coherent report:

The abnormality has a mass. The abnormality has a microlobulated margin. The abnormality has a round shape. The abnormality has a probably malignant assessment.

4 Robustness Evaluation

The robustness of the MIAKT generator was evaluated by running it on 350 different cases without modifications in between the runs. None of these 350 cases were used during the customisation of the ontology-based generator to the MIAKT domain, i.e., this was unseen data.

The experiment showed that there are no missing lexicalisations in the MIAKT terminological lexicon. Also, the MIAKT generator successfully produced reports for all 350 cases. There were some problems with the consistency of the RDF input in some cases, but the MIAKT generator proved robust in face of such noisy input.

The most frequent problem was that some RDF descriptions were incomplete. In these circumstances this information was not included in the report. For example, some RDF files would list 4 images as the result of a mammography exam (see Figure 2), but would not provide corresponding <Description> entries for them further down defining these instances, so the generator cannot refer to their location or their properties. For such instances without definitions the system currently chooses to omit them completely, e.g., the corresponding report in Figure 3 mentions only one image, rather than 4, because the other 3 instances are undefined. A warning message is produced in the GATE Messages tab alerting the user to the detected problems with the input.

In terms of performance, the generation of one report takes, on average, less than 20 milliseconds on a PIII 800MHz computer with 512MB memory, running Windows 2000 and Java as part of the JBuilder development environment. Batch

processing in a command prompt and without a development environment is some 10% faster. The average length of reports on these 350 test cases is 10 sentences. Performance evaluations of HYLITE+ on longer texts have shown comparable results [13].

5 Domain Portability

The ontology-based generator relies on a lexicon for all concepts and instances in the ontology and also on lexicalisations for all active action and passive action properties. In MIAKT they were all acquired manually and part of the porting effort to a new domain lies in the creation of these resources. However, this process can be facilitated substantially by importing information from existing resources such as UMLS and its SPECIALIST lexicon [6].

A new domain also means a different ontology. The main requirement towards the input ontology is to provide the hierarchical property structure discussed in Section 3.1. Further refinement of the `part-whole-property` into finer grained types such as `member-of-property` and `made-of-property` will increase the generator's expressiveness, but is not required.

If the ontology does not already have such a hierarchy, then it might be possible to create one semi-automatically, if there are well-defined naming conventions. For example, the UMLS Semantic Net [6] provides 54 links⁴ classified into a 3-level hierarchy with 5 top categories – `Spatially RelatedTo`, `Conceptually RelatedTo`, `Physically RelatedTo`, `Functionally RelatedTo`, and `Temporally RelatedTo`. Sub-links of each of these 5 top-level links tend map to at least 2 of our 4 generic properties. For instance, there are 4 links under `Spatially RelatedTo` – `Has_location`, `Adjacent_to`, `Surrounded_by`, and `Traversed_by`. From NLG perspective the first one is attributive, the second – an active action, and the third and fourth – passive action properties. Since UMLS links are named following certain conventions, it is possible to implement heuristics automatically mapping links with names starting with `has_` as subproperties of `attribute-property` and those ending with `ed_by` as subproperties of `passive-action-property`.

As discussed in Section 3.3, the patterns used for text structuring are typically domain and application specific. Therefore, the main effort in porting the generator to a new domain is in defining and implementing these patterns. Typically this involves the creation of a corpus of target texts and its analysis by a language generation expert, unlike the lexicon building and ontology modification tasks which can be carried out by knowledge engineers.

Recent work in language generation has started experimenting with machine learning to induce text structuring patterns from example texts annotated with the underlying semantic propositions. So far, only small semantically annotated corpora have been created because semantic annotation is time consuming since it requires a high level of detail, i.e., first annotating concepts and then the

⁴ Links in UMLS Semantic Net are equivalent to properties in Semantic Web ontologies.

relations between them. For example, [14] have collected an annotated corpus of 24 transcripts of medical briefings. They use 29 categories to classify the 200 tags used in their tagset. Each transcript had an average of 33 tags with some tags being much more frequent than others. Since the tags need to convey the semantics of the text units, they are highly domain specific, which means that this training data is specific to their application and domain and any model learned from this corpus will not be applicable in other circumstances. Future work on language generation in MIAKT aims at investigating how to lower the cost of adapting the discourse structuring component to new domains.

6 Related Work

NLG systems that are specifically targeted towards Semantic Web ontologies have started to emerge only recently. For example, there are some general purpose ontology verbalisers for RDF and DAML+OIL [10] and OWL [15]. They are template-based and follow closely the ontology constructs, e.g., *"This is a description of John Smith identified by http://...His given name is John..."* [15]. The advantages of Wilcock's approach is that it is fully automatic and does not require a lexicon. In contrast, the MIAKT approach requires some manual input (lexicons and domain schemas), but on the other hand it generates more fluent reports, oriented towards end-users, not ontology builders.

On the other end of the spectrum are sophisticated NLG systems such as TAILOR [16], MIGRAINE [17], and STOP [18] which offer tailored output based on user/patient models. In MIAKT we adopted a simpler approach, exploring generalities in the domain ontology, because our goal was to lower the effort for customising the system to new domains. Sophisticated systems, while offering more flexibility and expressiveness, are difficult to adapt by non-NLG experts. Our experience in MIAKT showed that knowledge management and Semantic Web ontologies tend to evolve over time, so it is essential to have an easy-to-maintain NLG approach.

The ONTOGENERATION project [2] explored the use of a linguistically oriented ontology (the Generalised Upper Model (GUM) [11]) as an abstraction between generators and their domain knowledge base. The project developed a Spanish generator using systemic grammars and KPML [19]. The main difference from our approach comes from the number of concepts and relations used to abstract the generator from the concrete domain ontology. In MIAKT we chose only 4 basic properties, in order to make it easier for non-linguists to carry out this task. The size and complexity of GUM make this process more difficult for non-experts. In general, there is a trade-off between expressivity and the number of linguistic constructs in the ontology. Therefore our approach is mainly suitable for applications where more schematic texts are sufficient and the goal is to have non-linguists being able to customise the generator for new domains.

This work also bears similarities with the problem of building portable and customisable NLG systems from relational databases [20]. Both our and ILEX approaches require a formal definition of domain knowledge as taxonomy or ontology and a mapping of ontology items to their lexicalisations. In the case of

Semantic Web ontologies, the information about domain types and data types of the slot fillers is already formally specified, unlike in databases. Our approach differs from that in [20] with its use of reasoning and a property hierarchy to avoid repetitions, enable more generic text schemas, and perform aggregation. Work on ILEX is complementary because it focused on low-cost methods for providing adaptivity and generation of comparisons.

7 Conclusion

This paper presented an approach for automatic generation of reports from domain ontologies encoded in Semantic Web standards like OWL. The novel aspects of the MIAKT generator are in the use of the ontology, mainly the property hierarchy, in order to make it easier to connect a generator to a new domain ontology. It also comes with a number of user-friendly tools for providing lexicalisations for the concepts and properties in the ontology [8], thus making it easier for non-specialists to customise a generator to their application. Our main contribution is in showing how existing NLG tools can be adapted to take Semantic Web ontologies as their input, in a way which minimises the portability effort while offering better output than template-based ontology verbalisers (e.g., [15]).

The system is still under development⁵ and is about to undergo user-based evaluation where medics will be asked to provide qualitative feedback on the readability and utility of generated reports. Preliminary feedback from the medical researchers involved in the project has indicated that such reports are perceived as well structured and understandable.

Future work will also aim to address the problem of generating tailored reports, depending on the user and the context. In the MIAKT domain the application required that all the information from the ontology about a given patient is included in the generated report. In other applications this might lead to overly verbose reports and thus methods for selecting only part of the available information will be required.

References

1. Reiter, E., Dale, R.: Building Natural Language Generation Systems. *Journal of Natural Language Engineering* **Vol. 3 Part 1** (1999)
2. Aguado, G., Bañón, A., Bateman, J.A., Bernardos, S., Fernández, M., Gómez-Pérez, A., Nieto, E., Olalla, A., Plaza, R., Sánchez, A.: ONTOGENERATION: Reusing domain and linguistic ontologies for Spanish text generation. In: *Workshop on Applications of Ontologies and Problem Solving Methods*, ECAI'98. (1998)
3. Hu, B., Dasmahapatra, S., Shadbolt, N.: From Lexicon To Mammographic Ontology: Experiences and Lessons. In Calvanese, D., De Giacomo, G., Franconi, E., eds.: *Proceedings of the International Workshop on Description Logics (DL'2003)*. (2003) 229–233

⁵ It is available online as a web service. To obtain a demonstration client or further information contact the first author.

4. Horrocks, I., van Harmelen, F.: Reference Description of the DAML+OIL (March 2001) Ontology Markup Language. Technical report (2001) <http://www.daml.org/2001/03/reference.html>.
5. Lassila, O., Swick, R.: Resource Description Framework (RDF) Model and Syntax Specification. Technical Report 19990222, W3C Consortium, <http://www.w3.org/TR/REC-rdf-syntax/> (1999)
6. NLM: Unified Medical Language System (UMLS). Technical report, National Library of Medicine, (<http://www.nlm.nih.gov/research/umls/umlsmain.html>)
7. Bontcheva, K., Kiryakov, A., Cunningham, H., Popov, B., Dimitrov, M.: Semantic web enabled, open source language technology. In: EACL workshop on Language Technology and the Semantic Web: NLP and XML, Budapest, Hungary (2003)
8. Bontcheva, K.: Open-source Tools for Creation, Maintenance, and Storage of Lexical Resources for Language Generation from Ontologies. In: Proceedings of 4th Language Resources and Evaluation Conference (LREC'04). (2004)
9. Bontcheva, K.: Generation of multilingual explanations from conceptual graphs. In Mitkov, R., Nicolov, N., eds.: Recent Advances in Natural Language Processing: Selected Papers from RANLP'95. Volume 136 of Current Issues in Linguistic Theory (CILT). John Benjamins, Amsterdam/Philadelphia (1997) 365 – 376
10. Wilcock, G., Jokinen, K.: Generating Responses and Explanations from RDF/XML and DAML+OIL. In: Knowledge and Reasoning in Practical Dialogue Systems, IJCAI-2003, Acapulco (2003) 58–63
11. Bateman, J.A., Magnini, B., Fabris, G.: The Generalized Upper Model Knowledge Base: Organization and Use. In: Towards Very Large Knowledge Bases. (1995) 60–72
12. Reape, M., Mellish, C.: Just what *is* aggregation anyway? In: Proceedings of the European Workshop on Natural Language Generation (EWNLG'99), Toulouse, France (1999) 20 – 29
13. Bontcheva, K., Dimitrova, V.: Examining the Use of Conceptual Graphs in Adaptive Web-Based Systems that Aid Terminology Learning. International Journal on Artificial Intelligence Tools – Special issue on AI Techniques in Web-Based Educational Systems (2004) Forthcoming.
14. Duboue, P.A., McKeown, K.R.: Empirically estimating order constraints for content planning in generation. In: Proceedings of ACL-EACL, Toulouse (2001)
15. Wilcock, G.: Talking OWLs: Towards an Ontology Verbalizer. In: Human Language Technology for the Semantic Web and Web Services, ISWC'03, Sanibel Island, Florida (2003) 109–112
16. Paris, C.L.: Tailoring object descriptions to the user's level of expertise. Computational Linguistics **14** (3) (1988) 64–78 Special Issue on User Modelling.
17. Mittal, V.O., Carenini, G., Moore, J.D.: Generating Patient Specific Explanations in Migraine. In: Proceedings of the Eighteenth Annual Symposium on Computer Applications in Medical Care, McGraw-Hill Inc. (1994)
18. Reiter, E., Robertson, R., Osman, L.: Lessons from a Failure: Generating Tailored Smoking Cessation Letters. Artificial Intelligence **144** (2003) 41–58
19. Bateman, J.A.: Enabling technology for multilingual natural language generation: the kpml development environment. Journal of Natural Language Engineering **3** (1997) 15 – 55
20. O'Donnell, M., Knott, A., Oberlander, J., Mellish, C.: Optimising text quality in generation from relational databases. In: Proceedings of the International Natural Language Generation Conference (INLG'00). (2000) 133–140

A Flexible Workbench for Document Analysis and Text Mining

Jon Atle Gulla, Terje Brasethvik, and Harald Kaada

Norwegian University of Science and Technology
Trondheim, Norway
jag@idi.ntnu.no

Abstract. Document analysis and text mining techniques are used to pre-process documents in information retrieval systems, to extract concepts in ontology construction processes, and to discover and classify knowledge along several dimensions. In most cases it is not obvious how the techniques should be configured and combined, and it is a time-consuming process to set up and test various combinations of techniques. In this paper, we present a workbench that makes it easy to plug in new document analysis and text mining techniques and experiment with different constellations of techniques. We explain the architecture of the workbench and show how the workbench has been used to extract ontological concepts and relationships for a document collection published by the Norwegian Center for Medical Informatics.

1 Introduction

The amount of document data in modern companies is growing rapidly. The rate of growth, combined with the documents' notorious lack of structure, makes it increasingly difficult both to locate the right documents and explore the information hidden in the documents. Research in information retrieval, document categorization, information extraction and ontologies tries to cope with these challenges. Even though their approaches are different, they tend to rely on many of the same underlying document analysis (DA) and text mining (TM) techniques.

These techniques analyze various properties of one or more textual documents using statistical and/or linguistic means. They range from simple counting of word frequencies to advanced semantic analyses of document content. Some of the techniques have already been introduced in commercial search engines [8], but most of them are still at a research stage and we do not know the full potential of them.

Typical techniques tend to analyze a stream of text and add some new piece of information. It can be a language detection technique that adds a language flag to a document or a tagger that adds a part-of-speech tag to every word in the document. A frequency technique may add the normalized frequency for each word to the document. However, some of the techniques depend on other techniques, which means that they have to be run in a certain sequence. A lemmatizer would normally require that the document is already tagged, and the tagger needs to know the language of the

document to work. The techniques, thus, have similar structures and need to be combined in particular orders to give the best possible result.

To support the use of statistical and linguistic text processing techniques, a number of tools have been developed. However, most current implementations of linguistic analysis tools and text processing techniques tend to be standalone programs, small-scale scripts, or parts of large and closed environments. They are created to address a restricted set of tasks, the results are kept internally, and there is little support for extending the analysis. Open architectures or data formats are rare to come across. The "small scripts" approach can be quite flexible, but leaves the user with the burden of managing all input/output, running each script in the proper order and configuring it all to fit together.

In this paper, we propose a *linguistic workbench* architecture that is open, flexible and expandable, allows new linguistic and statistical techniques to be easily added and configured, and simplifies the definition, execution, and application of DA and TM analyses. The next section of the paper describes the overall principles and architecture of our linguistic workbench. We show how the workbench is used to set up document analyses in Section 3 and go through an ontology construction case in Section 4. Whereas a discussion of related work is found in section 5, section 6 concludes the paper.

2 The Linguistic Workbench

The idea of this workbench was to provide a tool that is both simple to use and easy to expand as analysis needs grow. The tool should not require the presence of other applications and must be able to make use of already available text processing techniques. Programming or knowledge of particular systems should not be needed. The result is a tool set that includes a small web controller and a number of web services that can be combined to analyze the text of a given document collection.

Each document analysis and text mining technique is implemented as a *component* that reads a textual input file and transforms the content or expands it with new information. Some parameters may be used to control the behavior of the component. Several components are linked together in *jobs*, which define complete analyses of the document collection at hand. They consist of an input document collection and a sequence of configured components that take as input the result produced by the preceding component. After each component in the job, a temporary result file is generated and stored for later inspection.

To register new components, define jobs and configure the included components, a web controller is implemented. The controller calls the components in the correct order with the correct input files and make sure that both the job itself and all results are stored for later use.

Figure 1 shows the general structure of the workbench. The user selects the components to use from a library of available components. After indicating where the document collection is and configuring and ordering the components, the controller executes the job and generates a result file as well as a file for each component that was executed.

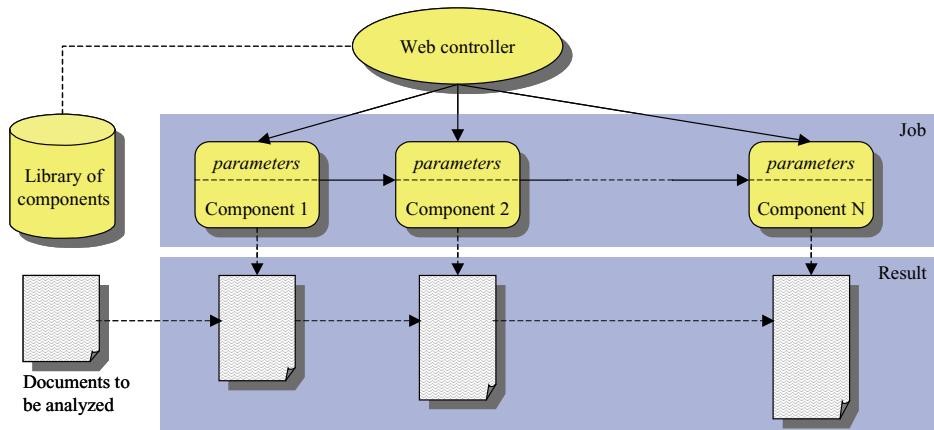


Fig. 1. Linguistic workbench structure

2.1 Architecture

Users define and configure jobs in a regular web browser. A controller, which is implemented in Python, stores and manages the configuration of jobs in a local MySQL database. Each component that is included in the job is referred to as a *job step*. Having defined the complete job with all necessary job steps, the user can execute the components of the job through an XML-RPC interface.

The components can be implemented in any programming language and can be executed on any networked computer, as long as the XML-RPC interface is supported. They must contain a standard interface for describing its possible parameters, indicating dependencies to other components (e.g. that phrase detection is dependent on the part-of-speech tagging) and receiving parameter settings and specifications of input and output. A component template is defined for this purpose. The components available so far are implemented in Python, Perl and Java. They run as web services that communicate and exchange information with the controller.

The document collections to be analyzed are stored as regular text files that must be accessible through the file system. The results are stored in an XML file in a format called *DOXML*. This file format is set up so that all results from job steps can be stored sequentially and the components can add information to the file by building on the previous results. Figure 2 shows a small part of a DOXML file from an analysis of Norwegian documents in the medical domain. It shows how the tagger, phrase recognizer and lemmatizer analyze the noun phrase *kliniske undersøkelser* (clinical examinations) and sequentially add information about part-of-speech, phrases and stems. Whereas *kliniske* is found to be an adjective, *undersøkelser* is recognized as a noun. The phrase is in plural, but the lemmatizer adds the singular nominative forms as base forms under the STEM tag. Different parts of the DOXML file, thus, are generated by different components, and the final file is a structured representation of the results from all the executed components.

The verboseness of the XML structure combined with the size of a document collection implies that the DOXML result file becomes quite large. As a result of this, all analysis of the file is performed sequentially. Since it has been impossible to keep the entire data structure of the result in memory simultaneously, the DOXML file is parsed using an event driven SAX parser rather than a DOM parser (w3c-xml-dom).

2.2 Workbench Components

Currently, the following components are implemented and added to the workbench library (see [11] for details):

Extraction: The extractor reads the entire document collection and constructs a DOXML document for each plain text or HTML document. Unwanted text like HTML tags are removed, meta-data is reorganized according to the DOXML format, and information about sentence borders, paragraphs borders and chapter borders are added. The extractor may include the text in its original form or only keep the text rearranged according to the rules of DOXML.

Language detection: The current language detection component only distinguishes between Norwegian and English and is based on standard text categorization techniques (see [3,5]). Dictionaries of the most frequent terms of each language is maintained and used to calculate a language score for the text. A parameter decides if the language should be identified at the level of sentences, paragraphs or chapters.

Text cleaning: This component makes sure that the document text is specified in the UTF-8 encoding and performs transliteration of special language characters. All accents, umlauts (e.g. ü) and diaeses are removed from the text. Hyphens within words and all variants of parentheses are also removed. For example, "münchen" is reduced to "munchen" and "state-of-the-art" ends up as "state of the art".

Part-of-speech tagger: The POS tagger assigns a part-of-speech flag to each word in a sentence. Our tagger is a stochastic Hidden Markow model based tagger using the Viterbi algorithm (see for example [10]). It uses a rather coarse-grained tagset for the 11 most common Norwegian word classes. The tags are KONJ (conjunction), TALL (number), DET (determiner), PREP (preposition and predeterminer), ADJ (adjectives), HJVERB (modal verb), SUBST (proper and common nouns), PRON (personal and possessive pronouns), ADV (adverbs), INF_MERKE (infinitive mark), and VERB (verb). A rather small tagged corpus (41,000 tokens) is used to compute bigrams for training the Markov model. If an unknown word is encountered, its tag is decided by comparing its ending with a list of suffixes and a list of inflectional endings or it is selected on the basis of the general probability of different word classes. Figure 2 shows how the tags are added to the document text.

Lemmatizer: The lemmatization component assigns a base form (dictionary entry) to each word in the document. The base form of *written* is *write*, and the base form of *better* is *good*. If a word has several possible base forms, like *searching*, the component uses the POS tag to decide which one to use. The word *searching* is expanded with the base form *searching* if it is a noun, but with the base form *search* if it is a verb. Noun phrases are also lemmatized, as shown in Figure 2. This is simply done by lemmatizing each word in the phrase independently of each other.

<pre> <W POS="3" V="kliniske"> <TAG>adj</TAG> <STEM>klinisk</STEM> </W> <W POS="4" V="undersøkelser"> <TAG>subst</TAG> <STEM>undersøkelse</STEM> </W> ... <PH> <NP POS="3" V="kliniske undersøkelser"> <NPSTEM>klinisk undersøkelse</NPSTEM> </NP> </PH></pre>	<p>Word "kliniske" (clinical) at position 3 Part of speech is adjective Stem of word is "klinisk"</p> <p>Word "undersøkelser" (examinations) at position 4 Part of speech is noun Stem of word is "undersøkelse"</p> <p>Noun phrase (adj+noun) detected at position 3 Stem of noun phrase</p>
--	--

Fig. 2. Text expanded with POS tags, base forms and phrase tags in DOXML

Stopword removal: The component removes words according to two strategies: (1) Remove all words that are on a pre-specified stopword list. These are typically very common words, like *in* and *is* in English, that are so frequent that they have limited discriminating value. (2) Remove all words that belong to certain word classes. In our Norwegian analysis, for example, we remove all adverbs, pronouns, determiners and ordinals.

Phrase detection: This component requires that the text is tagged beforehand. It is used to recognize and specify all noun phrases and verb phrases in the text. For our analysis, we define a noun phrase as a sequence of zero or more adjectives followed by one or more nouns (pattern Adj*N+). Adverbs in noun phrases, like *in very large databases*, have so far been ignored. Figure 2 shows how the noun phrase *kliniske undersøkelser* (clinical examinations) is recognized and added to the DOXML result document. Verb phrases are only recorded for sentences that contain noun phrases and are later used to suggest names for relations between concepts.

Weirdness analysis: the weirdness component counts term and phrase occurrences and calculates a weirdness measure using Ahmads Weirdness coefficient [1] and a reference collection. Let D_{dm} be the collection to be analyzed, D_{gen} the reference collection, and i the word to be analyzed. The weirdness coefficient for word i is then defined as follows:

$$(\# \text{ occurrences of } i \text{ in } D_{dm} / \# \text{ items in } D_{dm}) / (\# \text{ occurrences of } i \text{ in } D_{gen} / \# \text{ items in } D_{gen})$$

Figure 3 shows how this information is added to our DOXML document. We can calculate the coefficient for both individual words and phrases. The phrase *klinisk undersøkelse* has an absolute frequency of 20, a normalized frequency of 0.000166 and a weirdness coefficient of 24.7. A high coefficient means that the word is unusually prominent in the document collection and a good candidate concept for the domain. If the word is not found in the reference collection at all, a weirdness coeffi-

cient of INFINITY is recorded. This may also be a good concept candidate, though the reason may also be misspellings or other disturbances.

Co-occurrence analysis: The correlation analysis detects co-occurrences of words and phrases within a document, paragraph or sentence. For each word or phrase, we calculate a vector that shows all its occurrences in all documents. The similarity between terms is given by the standard cosine similarity measure that is common in information retrieval systems [2]. For each term (word or phrase), the component lists the N most similar terms. Similar terms may be synonyms, but the reason may also be that there are strong semantic relations between the terms.

3 Using the Workbench

When an analysis is to be carried out, the user first has to define a job. A new job is created, and the user selects all the components that are needed to do this analysis. The components then have to be configured with the correct parameter values. Figure 4 shows the list of all available components (back window) as well as the two specific windows for configuring the stopword component. The parameter in this case is which word classes to delete from the document text. The lower right window specifies that TALL (numbers), SUBST (nouns), DET (determiners), KONJ (conjunctions) and ADJ (adjectives) are to be removed from the text. For each component that is selected, the user needs to set the corresponding parameters.

When all the components have been selected and configured, the user needs to verify that they will run in the right order. The job called *Ivers dokumentanalyse* (Iver's document analysis) in Figure 5 contains 5 job steps. In the first step, the DOXML document format is generated from the HTML files. The second step removes stopwords, and the third step identifies the language of the text. Step 4 assesses prominent terms using the weirdness test, before a simple counting component is used to calculate term frequencies. If the order is wrong, the user may use the small arrows on the right hand side to move components up or down the list. If the stopwords are language-dependent, for example, the user may need to move the language detection component up to identify the language before any stopwords are removed.

```
<rank class="weirdness" ref_coll="file:/home/a/5/brase/weirdness/doxml/nou.xml">
...
<rf w="klinisk undersøkelse" freq="20" nfreq="1.6603987E-4">24.721827</rf>
<rf w="helseorganisasjon" freq="86" nfreq="7.1397144E-4">Infinity</rf>
...
</rank>
```

Fig. 3. Weirdness coefficients for words and phrases

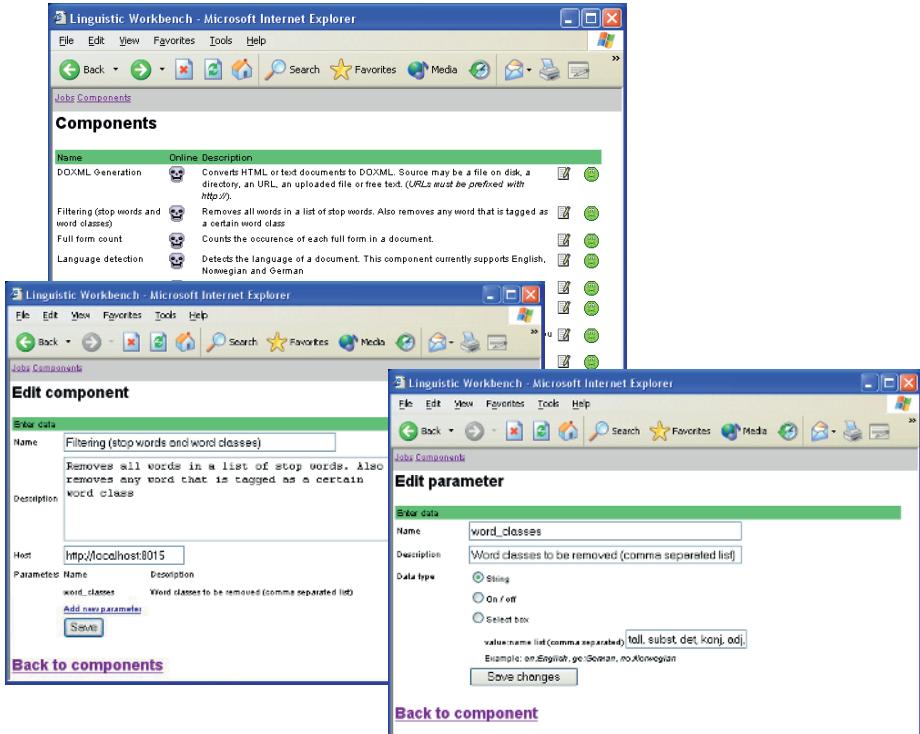


Fig. 4. Selecting and configuring a component

The result of the job is stored in a large DOXML file. This file can be directly inspected, and it will have the structure shown in Figure 2 and 3. However, it is not practical to display all information as an expansion of the original document text. A web query interface to some of the results has been set up to enable examination of temporary results, construct result subsets for further analysis and extract information from the result set that can be added to some domain model representation.

4 The KITH Concept Extraction Case

The Norwegian Center of Medical Informatics (KITH) has the editorial responsibility for creating and publishing ontologies for medical domains like physiotherapy, somatic hospitals, psychological healthcare, and general medical services. Traditionally, the ontologies are created on the basis of documents from the relevant domains and an elaborate modeling process. Committees of computer scientists, medical doctors and other stakeholders sit together and work out definitions of common terms and relationships between these terms. The result is a dictionary that lists all the relevant concepts, their definitions, and their relationships to other concepts (cross-references). The example below, which is taken from the *health school* ontology, presents one of the 110 concepts defined [12].

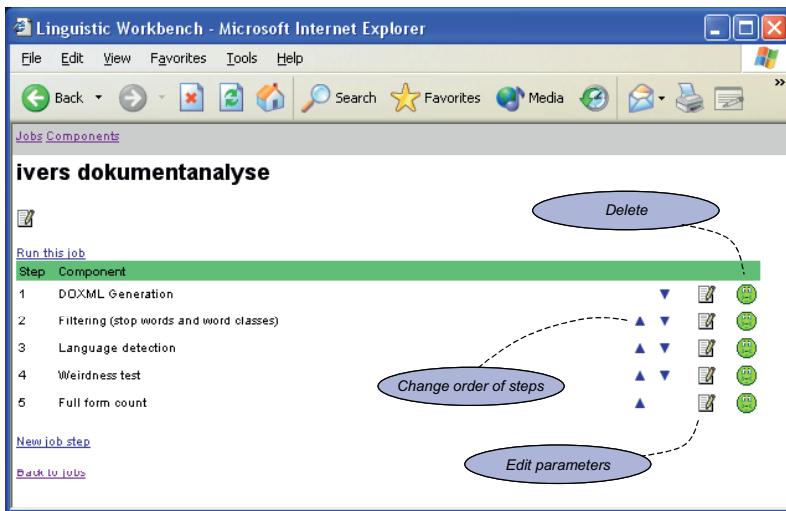


Fig. 5. The job steps of a job

No.	Term	Definition	Cross-reference
67	Patient	A person contacting the health services to request health care, or a person that is offered health care by the health services at some time	<i>Departed patient 100</i> <i>Day-care patient 17</i> <i>Guest 23</i> <i>Patient ready for departure 104</i> <i>Patient no. 70</i>

The objective of our case analysis was to extract and define concepts to be included in their ontology. After experimenting with different techniques and different combinations of techniques, we set up the analysis shown in Figure 6. The overall idea was to remove all syntactic variation and unimportant words from the text and then run frequency tests to pull out the prominent concepts and relations from the collection. Part of the analysis could be carried out by the workbench alone, but we also realized that there had to be some manual intervention when concepts and relations between concepts were selected.

The document collection used in this case was a 2.79 MB collection of documents concerning health and school issues. As a reference collection for the weirdness test, we used 39.3 MB of Norwegian public reports available from Odin (www.odin.no). An extraction component first removed the HTML tags and reorganized the content according to the structure of our DOXML format. Since both English and Norwegian text could appear in the documents, we needed to run the language detection component to tag each paragraph with the right language code. Some special characters like hyphens and accents were removed, before the tagger annotated all words with the appropriate word class. This information about word classes was used by the lemmatizer to add the base form of each word that appeared in the document. The POS tags were also used to remove unimportant word classes like determiners and adverbs and to identify noun phrases in the text. The weirdness component was run to identify concept candidates, where a concept is regarded as the base form of a word or phrase.

The frequencies, thus, were calculated from base forms rather than from the actual word forms. Figure 7 shows the top 10 concept candidates found for two particular cases. Whereas the candidates in (a) did not appear at all in the reference collection, the candidates in (b) were prominent in the sense that they appeared a lot more frequently in our collection than in the reference collection. Both lists included very good concept candidates that were already in KITH's manually constructed ontology.

A manual selection of concepts was then carried out. For each concept that was selected, the system collected all the inflectional variants of the words referring to this concept and added them to a concept lexicon. The workbench analyzed potential relationships between the concepts using the co-occurrence analysis component. Each concept was linked to a number of other concepts that tended to occur in the same context. We ran experiments with the context set to sentences, paragraphs or documents, though the results were quite consistent across these tests. The hypothesis was then that concepts that occur together should be semantically linked. If that hypothesis holds, the example in Figure 8 indicates that there may be important semantic relations between *mobbing* (mobbing) and the concepts of *sosial ferdighet* (social skills), *skole* (school), and *trivsel* (well-being). Most concept pairs identified by this component indicated valid semantic relations. To keep the resulting ontology clean and manageable, though, there was a question how many relations should be included at the end. The selection of these relations as well as their descriptions were left to the experts at the end of the whole experiment.

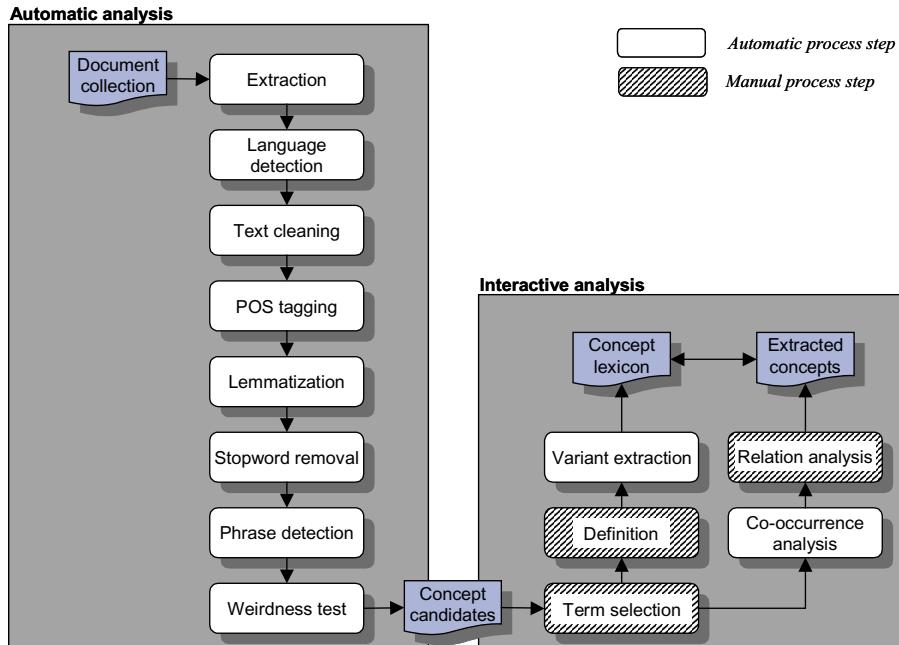


Fig. 6. The KITH concept extraction job

(a)	No.	Term	Frequency	Weirdness
	1	helsestasjons (health station's)	405	Infinity
	2	helseorganisasjon (health organization)	86	Infinity
	3	journalsystemet (journal system)	69	Infinity
	4	kvalitetsådgiverprogrammet (quality advisory program)	57	Infinity
	5	miljørettet (environmental)	56	Infinity
	6	journalopplysninger (journal information)	46	Infinity
	7	sped (infant)	40	Infinity
	8	helsekortet (health card)	34	Infinity
	9	skolehelsetenesta (school health service)	32	Infinity
	10	journalforskriften (journal regulation)	31	Infinity

(b)	No.	Term	Frequency	Weirdness
	1	skolehelsetjenesten (school health service)	618	8402.9
	2	pasientjournaler (patient records)	85	1155.7
	3	habilitering (rehabilitation)	80	1087.8
	4	fastlege (primary doctor)	54	734.2
	5	helsestasjon (health station)	366	497.7
	6	helseopplysning (health information)	87	394.3
	7	barn elev (child pupil)	29	394.3
	8	brukernavn (user name)	28	380.7
	9	spedbarn (baby)	24	326.3
	10	måltid (meal)	47	319.5

Fig. 7. (a) Top prominent concepts not found in reference collection. (b) Top prominent concepts found in reference collection.

Our approach to concept extraction was substantially faster than the process KITH had used in the past. The workbench found 99 of the 111 concepts in KITH's ontology, but the workbench also discovered concepts and relations that had not been recorded previously, but should be included in the ontology. Moreover, KITH intends to use the ontology to help the user search for documents. As our ontology accurately reflects the content of the document collection, it should be better for query reformulation and document navigation. KITH has already set up a successful search prototype using the results from our workbench (www.volven.no). An important advantage of the workbench was the flexibility it offered in setting up different hypotheses and analysis chains. We could experiment with different components and different ordering of components with little effort and with no technical expertise.

Term	No.	Related term	Similarity measure
mobbing (mobbing)	1	mobbing (mobbing)	1.0
	2	sosial ferdighet (social skills)	0.53884
	3	skole (school)	0.48495
	4	trivsel (well-being)	0.43244
	5	rusmiddel (drugs)	0.42925
	6	ernæring (nutrition)	0.42354
	7	selvmord (suicide)	0.40705
	8	ungdom (youth)	0.40344
	9	elev (pupil)	0.39500

Fig. 8. Relations between concepts

5 Related Work

Document analysis and text mining techniques are now used in many practical large-scale systems, like for example the FAST search engine on www.alltheweb.com [8]. There are also many experimental systems, in which innovative use of linguistic & text mining techniques are found (e.g. [4, 6, 9]).

The functionality of our workbench is comparable to what we find in some comprehensive ontology and concept extraction environments [13, 14, 15]. These tools are larger than our workbench and include techniques that are more sophisticated than the ones running on our workbench at the moment. Our workbench is somewhat different in the sense that it is a light-weight, application-independent and totally expandable tool. It is intended for people with limited knowledge of text mining that would like to run experiments with already implemented components. Also, it is designed to allow easy integration of components that have been written for other applications.

As pointed out in [7, 14], the results of text mining experiments are uncertain and unpredictable. One has to expect a whole series of experiments before the right combination of techniques is found for a particular text mining task. Our workbench is specifically designed to make it easy to rearrange text mining experiments and test out different combination of statistical and linguistic techniques. The DOXML format and the standard component interface make it possible to control all this with a simple graphical web interface.

We have in this paper presented a flexible and expandable workbench that is used to integrate and coordinate the use of document analysis and text mining techniques. While the components may be implemented in different languages and running on different computers, the users compose their own analyses with a simple web interface. A number of linguistic and statistical components have already been added, and new ones can easily be introduced as the needs arise. To make the workbench as applicable and predictable as possible, we focus on components that are well described and defined in the research community (see for example [3, 5]).

The experiment on concept extraction for KITH reveals that the workbench speeds up the process of constructing ontologies from document collections. It also shows that the simplicity of the tool does not prevent it from running real text mining analyses and producing high quality results. However, it seems difficult to fully automate the concept extraction process. Even though the candidate concepts and candidate relations were reasonable, the experts needed to go through the list and verify the suggestions from the workbench.

At the moment, most information generated by the workbench is added to a DOXML document that grows bigger for every component that is executed. This is needed for some of the linguistic techniques, though the format may not be ideal for some of the statistical techniques. The co-occurrence analysis component, for example, tells us which concept pairs are most likely to be semantically related. These results are better stored in a tabular format that can be queried by users or other components. A standardized approach to storing these kinds of results remain to be defined and implemented as part of the workbench.

References

1. Ahmad, K. (1994) "Language Engineering and the Processing of Specialist Terminology". [<http://www.computing.survey.ac.uk/ai/pointer/paris.html>].
2. Baeza-Yates, R. and B. Ribeiro-Net (1999). *Modern Information Retrieval*. Addison Wesley Longham.
3. Berry, M. W. (2004). *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer.
4. Brasethvik, T. and J. A. Gulla (2001). "Natural Language Analysis for Semantic Document Modeling." *Data & Knowledge Engineering*. Vol. 38, No. 1, July 2001, pp. 45-62.
5. Chakrabarti, S. (2003). *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers.
6. Desmontils, E., and C. Jacquin (2001) "Indexing a Web Site with a Terminology Oriented Ontology". In *Proceedings of the first Semantic Web Working Symposium (SWWS'01)*, pp. 549-565. Stanford, July/August 2001.
7. Faure,D. and T. Poibeau (2000) "First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX". In *Proceedings of the First Workshop on Ontology Learning OL'2000*. ECAI Workshop on Ontology Learning. Berlin, August 2000.
8. Gulla, J. A., P. G. Auran, and K. M. Risvik (2002). "Linguistics in Large-Scale Web Search". In *Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems (NLDB 2002)*, Stockholm, June 2002, pp. 218-222.
9. Haddad, H. (2002). "Combining Text Mining and NLP for Information Retrieval". In *Proceedings of the International Conference on Artificial Intelligence (IC-AI '02)*, Vol. 1, June 2002, Las Vegas, pp. 434-439.
10. Jurafsky, D. and J. H. Martin (2000). *Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.
11. Kaada, H. (2002) "Linguistic Workbench for Document Analysis and Text Data Mining." Master's thesis. Norwegian University of Science and Technology, Trondheim.
12. KITH (2003). *Definisjonskatalog for helsestasjoner og skolehelsetjenesten*. KITH report 15/02. ISBN 82-7846-140-6, First edition. In Norwegian.
13. Kunze, M. and D. Rösner (2001). "An XML-based Approach for the Presentation and Exploitation of Extracted Information." In *Proceedings of the 1st International Workshop on Web Document Analysis (WDA'2001)*, September 2001, Seattle.
14. Maedche, A. and S. Staab (2001). "Ontology Learning for the Semantic Web". *IEEE Intelligent Systems*, March/April 2001, pp. 72-79.
15. Navigli, R., P. Velardi, and A. Gangemi (2003). "Ontology Learning and Its Application to Automated Terminology Translation". *IEEE Intelligent Systems*, January/February 2003, pp. 22-31.

Towards Linguistic Foundations of Content Management

Gunar Fiedler and Bernhard Thalheim

Computer Science and Applied Mathematics Institute, University Kiel,
Olshausenstrasse 40, 24098 Kiel, Germany
{fiedler|thalheim}@is.informatik.uni-kiel.de

Abstract. Content and content management have become buzzwords. The notions are neither well-defined nor used in a standard way. Content objects are complex structured. Different users may require different content object sets. Content object sets may vary depending on the actual task portfolio, depending on the context of the user, and on the technical environment. Therefore, content management must combine generation, extraction and storage of complex object, must support complex workflows and must be adaptable to the actual use and users environment and requirements. Content may be considered under three different points of view: data computed from an information system, general concepts that are illustrated or described by the content, and, finally, a large variety of user interpretation. Since all three points of view should not be mixed with each other we propose to separate them and treat content management from the point of view of syntax, from the point of view of semantics and from the point of view of user worlds.

1 Content and Content Management

Content management, simply stated, is the process of sharing information vital to an organization. Likewise, intranet content management involves sharing information using the private computer networks and associated software of intranets (or extranets) as a primary communication tool. In today's "information society", where the total quantity of data and the pace of communication continue to increase, the goal of effective content management continues to gain importance.

Roughly we may classify content management systems into website content management systems, enterprise content management systems, advanced document management systems, and extranet content management systems. This large variety of systems has a number of properties in common: generation, delivery and storage of complex structured objects; rights management; service management in distributed environment; customer management; update and quality management; context dependent delivery depending on the user, the HCI, and the actual system situation.

The content of a CMS is a most value asset. Content must be updated frequently to keep user coming back and to succeed in their tasks. Thus, a content

management system supports production of content while automating some of the frequent operational tasks.

CMS and web CMS specifically support a variety of tasks like managing web assets from different sources, workflows, templates for content or presentation, source control and versioning, deployment and delivery of content as well as managing of distribution and customer adaptation.

Therefore, we claim that CMS must

- integrate extraction, storage and delivery of complex structured objects,
- support workflows and tasks,
- be based on service systems, and
- deliver content objects to users on demand and profile, at the right moment, and within the right format and size.

2 Semiotic Separation of Information into Content, Concepts, and Topics

Content is often considered to be a generalization of knowledge, information, and data. This generalization must capture all aspects of concern. Instead we prefer a separation of aspects of concern:

Pragmatics concentrates on the meaning of utterances.

Semantics expresses the interpretation of utterances used in a language.

Syntax restricts attention to the language, its construction, and the way of using it through utterances.

This separation is expressed in the semiotic triangle in Fig. 1.

We may distinguish information and content. Information as processed by humans, is *data* perceived or noticed, selected and organized by its receiver, because of his subjective human interests, originating from his instincts, feelings, and experiences simultaneously processed by his cognitive and mental processes, and seamlessly integrated in his recallable knowledge. Information is content

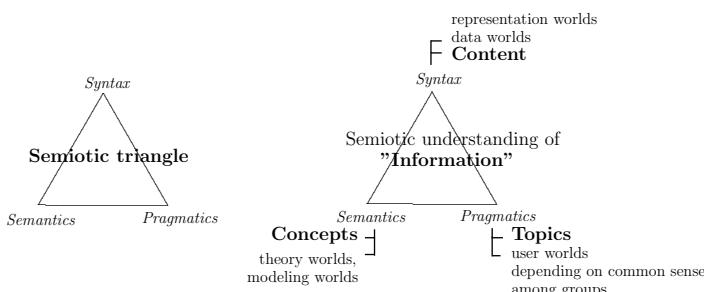


Fig. 1. Linguistic separation of concern applied to content theory

combined with the interpretation and the understanding of its users. We can now use this separation of aspects of concern to develop a semiotic understanding of concepts and information. This separation is displayed in Fig. 1. We use the word "content" in a restricted form and mean elements or subsets of business data. The word "concept" is used for theoretically based modeling worlds. It is thus used within the semantic or logical theory that forms the basement for concept worlds. Content and concepts may be denoted in a large variety by different users. Each user has his/her own terminology. We use the word topic for denotation of concepts or content. Therefore we distinguish between three different worlds of users:

Considering the **content world** or **representation world** we concentrate our attention to the data and their representation.

Considering the **concept world** we concentrate our investigation to the logical foundations of the content world and of the topic world.

Considering the **topic world** we are interested in the terminology or in the ontology of users, their common understanding.

There are tight relationships between these worlds. Users have their own understanding of their topics and may associate different content with the same topic.

Since we are targeting in a general specification framework for content systems we will not deepen the association to concepts and topics. Defining this theory of CMS we need to solve a number of additional problems: the definition of content, concepts and topics, the development of mappings between them, and the development of a supporting infrastructure, an integrated functionality and a modeling language.

In the next sections we develop a program for the solution of these problems.

3 Definition of Concepts

Concepts \mathfrak{C} are described by the triple

(meta information, intension specification, extension) .

The intension can be specified by providing the logical theory on the basis of a set of formulas of the predicate logics. The extension of \mathfrak{C} specifies the mappings to the content spaces and is used for associating content with concepts.

The **concept intension** is given by the **intext** of the concept (syntactical description of mandatory, normal and optional parameters), by the **context** of the application area, by **semantics** specified over the intext and the context, and by **usage** and **pragmatics** that restricts the application and the usage of the concept.

Since concepts are rather 'small' logical theories we use **concept nets** for the specification. Concept nets are specified by instantiating a concept definition frame. Concept fields generalize word fields used in [1].

Meta-information is used for specification of quality, restrictions of usage, ownership, replacement of concepts, and other associations to concepts. **Concept extension** is used for specification of the concept world. Since concept intensions

are ‘small’ logical theories we use logical models of concept intensions for concept extensions.

In contrast to common synonym dictionary, word fields define the possible/necessary actors, the actions and the context. Word fields can be used for verbs, nouns and adjectives. We focus on verb fields and extend the implementations of the *WordNet* dictionary for verbs. For each verb field we can at least define the basic semen, the context of usage, possible/necessary actors, actions and context, as well as the star or snowflake specification. Word fields are well-specified for a number of verbs.

In our generalization, concept fields are different from word forms that carry the grammatical variants. A word is an abstract concept which is concretely manifested solely in the associated word forms [5]. Word fields combine aspects defined in phonology (sounds), morphology (word form), syntax (sentence), semantics (meaning) and pragmatics (use).

4 Definition of Topics

Our topic notion generalizes and formalizes the notion of topics [8] commonly used for topic maps and implemented in [6]. Topic maps are based on conceptual structures [11]. Our notion integrates these proposals with the Pawlak information model [7] and concept lattices [4].

Topics \mathfrak{T} are described by the triple

(user characterization, topic description, topic population) .

User characterization may be based on the specification of users knowledge or beliefs, on the personality profile, and on the task portfolio. Topic description \mathfrak{T}_D can be given by the specification of

(topicRef, subjectIdentity, scope, baseName, association, roles, member, parameters).

Topic population \mathfrak{T}_P is specified by the specification frame

(instanceOf, resourceRef, ressourceData, variant, occurrence).

We typically require that topic description and topic population are **coherent**. The coherency notion is based on the information model of Pawlak (see [12]).

5 Definition of Content

A **content system** consists of a content management system and a set of content object suites. A content object *suite* consists of a set of elements, an integration or association schema and obligations requiring maintenance of the association. In the case of a content suite, we specify content objects based on a type system enabling in describing structuring and functionality of content objects, in describing their associations through relationship types and constraints. The functionality of content objects is specified by a retrieval expression, the maintenance policy and a set of functions supporting the utilization of the content object and the content object suite.

Our notion extends modern approaches to content suites [10] and combines them with the theory of media objects [2]. Classically, (simple) views are defined as singleton types which data is collected from the database by some query.

A *content schema* \mathfrak{D} on a database schema \mathcal{S} consists of a view schema \mathfrak{D}_V , a defining query $q_{\mathfrak{D}}$, which transforms databases over \mathcal{S} into databases over \mathfrak{D}_V , and a set of functions defined on the algebra $\mathfrak{A}(\mathfrak{D})$ on the content schema.

The defining query may be expressed in any suitable query language, e.g. query algebra, logic or an SQL-variant. For our purposes, however, this is yet not sufficient. One key concept that is missing in the views is the one of *link*. Therefore, we must allow some kind of "objectification" of values in the query language. A set $\{v_1, \dots, v_m\}$ of values is transformed into a set $\{(u_1, v_1), \dots, (u_m, v_m)\}$ of pairs with new created URLs u_i of type *URL* – more precisely, we first get only surrogates for such URLs. In the same way we may objectify lists, i.e. transform a list $[v_1, \dots, v_m]$ of values into a list $[(u_1, v_1), \dots, (u_m, v_m)]$ of pairs. We shall talk of *query languages with create-facility*.

As a second extension we may want to provide also auxiliary content [3]. An *extended content type* has a name \mathfrak{D}^x and consists of a content schema \mathfrak{D} , a defining query $q_{\mathfrak{D}^x}$ with create-facility that defines a view, and a binding between \mathfrak{D} and $q_{\mathfrak{D}^x}$.

Auxiliary content types are not yet sufficient for information service modelling. The representation of content types and auxiliary content types may depend on the user profile, on the user task portfolio, on units of measures, on representation types, on the order of representation, and finally on the container for delivery [9]. Therefore, we use *wrapped content types* \mathfrak{D}^w for representation of content types that are adaptable to users and to the environment.

6 Layering of Content, Concepts, and Topics

Topic maps, concept worlds, content suites, and databases can be organized in layers. At the pragmatics layer people are interested in their topics or topic maps. These maps are associated to theories from the semantics layer. Theories are used to express the understanding through topics. Theories are interpretations of content, i.e., business data. Business data consist of content. Content may be obtained through querying from data and may be conceptualized to concepts. This layering structure follows classical abstraction concepts.

We observe, furthermore, that different quality criteria may be applied at different layers. For instance, at the pragmatics layer users are interested in utility and usability. Other properties are functionality and understandability. On the other hand, at the database layer, quality requirements such as performance, concurrency, recoverability, security, and portability are used for system evaluation.

7 Conclusion

This paper proposes a general theory of content management systems in the broader sense. Content management can be separated into content management based on advanced database systems, concept management, and topic management. We have shown in this paper how content management systems in the broader sense can be built. This approach has already got a partial "proof of concept" in some of our projects. The next step of our research is the development of a general theory of operations and integrity constraints for content management.

References

1. A. Düsterhöft and B. Thalheim. Integrating retrieval functionality in websites based on storyboard design and word fields. In *Proc. NLDB'02, LNCS 2553, Springer*, pages 52–63, 2002.
2. T. Feyer, O. Kao, K.-D. Schewe, and B. Thalheim. Design of data-intensive web-based information services. In *Proc. WISE 2000, Volume I (Main Program)*, pages 462–467, 2000.
3. T. Feyer, K.-D. Schewe, and B. Thalheim. Conceptual design and development of information services. In *Proc. ER'98, LNCS 1507, Springer*, 1998, pages 7–20. Springer, Berlin, 1998.
4. B. Ganter and R. Wille. *Formal concept analysis - Mathematical foundations*. Springer, Berlin, 1998.
5. R. Hausser. *Foundations of computational linguistics*. Springer, Berlin, 2000. in German.
6. <http://www.ontopia.net/topicmaps/>.
7. Z. Pawlak. Mathematical foundations of information retrieval. Technical Report CC PAS Reports 101, Warszawa, 1973.
8. S. Pepper and G. Moore (eds.). Xml topic maps (xtm) 1.0. <http://www.topicmaps.org/xtm/1.0/>, 2001. TopicMaps.Org.
9. K.-D. Schewe and B. Thalheim. Modeling interaction and media objects. In M. Bouzeghoub, Z. Kedad, and E. Métais, editors, *NLDB. Natural Language Processing and Information Systems, 5th Int. Conf. on Applications of Natural Language to Information Systems, NLDB 2000, Versailles, France, Jun 28-30, 2000, Revised Papers*, volume 1959 of *LNCS*, pages 313–324. Springer, 2001.
10. J.W. Schmidt and H.-W. Sehring. Conceptual content modeling and management - the rationale of an asset language. In *Proc. PSI'03, LNCS , Springer*, 2003, 2003. Perspectives of System Informatics.
11. John F. Sowa. *Knowledge Representation, Logical, Philosophical, and Computational Foundations*. Brooks/Cole, a division of Thomson Learning, Pacific Grove, California, 2000.
12. B. Thalheim. *Entity-relationship modeling – Foundations of database technology*. Springer, Berlin, 2000.
See also <http://www.informatik.tu-cottbus.de/~thalheim/HERM.htm>.

Constructing Natural Knowledge Ontologies to Implement Semantic Organizational Memory

Laura Campoy-Gomez

Information Systems Institute, University of Salford

Manchester M5 5WT, United Kingdom

l.campoy-gomez@salford.ac.uk

<http://www.isi.salford.ac.uk/staff/lcg>

Abstract. Ontologies have proven to be excellent tools in the modelling, capture and implementation of the so-called organizational memories within the Knowledge Management domain. In this paper, an ontology-based existing framework for cooperative and consensual knowledge construction is analyzed. In this framework, ontologies are used for the implementation and development of a corporate memory through ontology integration as the basis for knowledge construction. Moreover, collaborative agents participating in the knowledge construction process will perform the integration mechanism by request, and can benefit from the use of individual, personalized (ontological) knowledge. Certain limitations of this approach are pointed out, particularly in relation to the mechanisms for knowledge integration and personalization. Further extension of the framework is also addressed.

1 Introduction

The application of Knowledge Engineering techniques and tools into the broad area of Knowledge Management is not a new thing. Thus, amongst others, rule-based systems, problem solving methods (PSM) and more recently, ontologies are contemplated at present as elements with which to model and implement the fundamental processes implied in the management of knowledge: the acquisition, storage, use and dissemination of knowledge.

Furthermore, the nature of the knowledge that these conceptual devices (rules, PSM, ontologies, etc) can hold and handle would depend on the actual implementation and on the complexity of the framework in which they are used. For example, gradual knowledge rules have proven to be rather appropriate to express simple and undetermined variations for natural language qualifiers such as “the more... the less”, “the higher... the less”, etc applied to parameter values, as in [1]. Another example could be the inclusion of temporal relations (like “after” and “before”) in ontologies coexisting with other kind of relational links such as mereological or taxonomic, which can permit the capture and permanent storage of temporal information, as in [2]. That is not to say that particular relational qualifiers (or of other kind) such as the mentioned before cannot be expressed when the conceptual machinery employed (e.g. ontologies) in information systems architecture does not directly support them. It would be more difficult, nonetheless, than in cases when tools or representation lan-

guages include primitives with which those notions can directly expressed as in the case of the so mentioned temporal relations.

Amongst all knowledge technologies derived from the field of KE, ontologies are demonstrating to be the most promising tool for capturing, representing, sharing and disseminating knowledge fundamental processes for the management of knowledge. Different attempts of implementing ontology-based information systems for Knowledge Management (KM) can be found in literature see e.g. [3], [4], and [5].

In this work the interest focuses on the modelling and construction of organizational knowledge making use of ontologies, as it is introduced in next section. In Section 3, an existing framework for corporate memory construction and fundamental processes based upon ontologies will be described. Section 4 will present a discussion on the extent and consequences regarding the implementation of these fundamental processes. Finally, in Section 5 further enhancement of this framework is addressed, pointing to the adoption of semantic web technologies.

2 Ontologies to Construct Organizational Memories

The notion of organizational memory can be contemplated in two ways. Firstly, we can find in literature definitions that would envision an OM as a recipient that would gather all knowledge elements produced as the result of KM processes. For example, in [6] an OM is understood as an explicit and persistent representation of the knowledge and information residing in the organization. In a similar line, several authors contemplate what has been called organizational memory information system (OMIS), assuming the existence of a system (formal or already implemented) to support these processes. On the other hand, other approaches focus on the second underlying meaning of memory, that being the ability of organizations (or individuals) to retain knowledge and information on past events, in analogy to the idea of memory for computer systems.

In this paper, the first of these two meanings would be assumed, that is OM is seen as a container that gathers, stores, integrates and disseminates knowledge items in the form of concepts belonging to an ontology resulting from KM processes at organizational level. Our interest is in ontology-based models for OM. In particular, the following sections will describe and analyze an ontology-based model for the implementation of an OM, and will illustrate the role of ontologies as building blocks for the construction of OM.

3 A Framework for Cooperative Knowledge Construction

In [7] a framework for cooperative and distributed construction of OM in which ontologies are the fundamental elements was presented. The essential idea around which this formal framework has been built and conceived was the capacity of a well-defined, inconsistency-free knowledge integration process to enable different and distributed agents to generate and update newly constructed knowledge onto a common global OM. The nature of this model of OM is purely ontological, for each contributing agent (knowledge expert user) maintains his/her private ontology, which will

be an instantiated (personalized) vision of the global memory and may include additional recently added knowledge to be potentially included in the OM as a result of an integration request. The OM is thus modeled to be a global ontology, generated and maintained through integration of individual contributions throughout time. These contributions would correspond to the particular vocabulary of individual users (in terms of concepts, its attributes and its relations to other concepts) and would represent different visions of the same knowledge, i.e. their terminology. The main ideas implemented in this formal framework can be summarized as follows:

Redundancy prevention. The process of integration formally defined in this framework was designed not to permit any redundant knowledge to be added to the global ontology, as the knowledge contained in the OM was supposed to be consensual knowledge, given the organizational nature of the cooperative work. For this, the integration algorithm detects when portions of knowledge in two different ontologies may correspond (conceptually and/or structurally) to the same knowledge, taking only the portion that could add more knowledge to the OM, and discarding the other.

Inconsistency prevention. Another semantic quality that had to be formally defined and implemented within this formal framework has been inconsistency. Accordingly, different portions of knowledge will be checked for consistency – both at conceptual and at ontology level – as a tool in the integration process to discard any portion of knowledge – concepts or ontologies – that could result to be inconsistent with the same portion of the knowledge already in the OM. The principle stated in this case to discard inconsistent knowledge was to take the latest portion of knowledge, as the one knowledge resident in the OM may have been consulted (hence taken as reference) for the construction of the former portion, the one corresponding to a particular agent.

Synonymy support. Another characteristic of this framework is the consideration of synonymy at concept level to enable the necessary later personalization of ontologies to display individual, personalized visions of the knowledge in the OM, so that different agents could work with different terminologies.

The modus operandi of this formal framework - based upon the above principles - would construct a global OM ontology by gathering and contrasting all individual participating ontologies from individual users – knowledge constructors - both at concept and at ontology level. Firstly, only the larger set of user node ontologies that can be all ‘compatible’ will be chosen to participate in the knowledge integration process, where compatibility is defined to be true when applied to two ontologies if they are neither inconsistent nor equivalent. Formal functional definitions of these notions can be found in [7].

Basically, two ontologies will be considered to be equivalent if, for each concept belonging to the first one there can be found some other concept in the second one such that both have the same attributes and parent/children concepts. Two concepts will be equivalent if both have the same attributes and parent/children concepts yet different concept names.

The definition of inconsistency may seem less obvious. Two ontologies will be considered inconsistent if there are at least two concepts, one belonging to the first one and another one belonging to the second one such that one of these two condi-

tions hold: i) both concepts have the same name, not having any attributes in common, while their respective parent/children concepts (if any) do have the same attributes, or ii) both concepts have the same attributes, there is no other concept being parent of any of them and having the same attributes than any parent of the other concept; the same property holding for children concepts.

Finally, all remaining ontologies susceptible of being integrated would participate in a transformation process at ontological level that would merge attributes and relations for equivalent concepts, level by level starting from their root concepts, taking always the portion that could add more knowledge to the global memory such as the ontology with more concepts, or the concept having more inherited or specific attributes. The above principles and concepts are reflected in the following algorithm (adapted from [2]), to illustrate the modus operandi of this formal framework.

3.1 An Algorithm for Ontology Integration

The following lines illustrate the algorithm for the construction of the global OM ontology by gathering and contrasting all individual participating ontologies, both at concept and at ontology level.

```

Ontological_integration algorithm
{Let OM(t) be the OM global ontology derived from the integration
process;
Let Oi(t) be the i-th ontology to participate in the integration
process for the construction of the global ontology OM;
Let N be the total number of participating ontologies;
Let participants(t) be the set of ontologies to integrate;
All refer to instant t};
For i=1 to N
    If (there exists Oj(t) belonging to participants(t)
        such that both, Oj(t) and Oi(t), belong to same user)
        Then (remove the older of the two)
        End-if
    End-for
Subset=Select_Ontologies(participants(t))
i=1
While i≤ Cardinal(subset) do
    Ontological_Inclusion(Oi(t))
End-while
Ontological-Transformation(OM)
End-Ontological_Integration

```

For reasons of clarity, some steps have been grouped into three other sub-algorithms, namely Select_Ontologies(), Ontological_Inclusion(), and Ontological_Transformation, which will be expanded in the following lines.

```

Select_Ontologies sub-algorithm
{Let compatiblei(t) be the set of ontologies Oj(t) belonging to
participants(t) that are compatible with Oi(t);};
For i=1 to cardinal(participants())
  For j=1 to cardinal(participants())
    If (compatiblei(Oi(t), Oj(t))
      Then (compatiblei(t) = compatiblei(t) ∪ Oj(t))
  End-for
End-for
Return (subset with higher number of ontologies)
End-Select_Ontologies

```

The above algorithm would illustrate the process of selection of the more numerous possible subset of compatible ontologies amongst the ones susceptible for integration. Next would represent the process of inclusion of a (user node) ontology within the global OM ontology. The idea behind the later would be to gather all individual terminologies for later access to the memory.

```

Ontological_inclusion sub-algorithm
{Let Oj(t) be the j-th ontology susceptible of being incorporated
into OM; Let topic-concept be the topic on which corresponding user
is generating knew knowledge about; Let Oi(t) be the ontology whose
root is topic-concept-i, i.e., topic according to ith user in OM;};
Add Oj(t) to OM as mereological child concept,
Root(Oj(t))=topic-concept-j
End- Ontological_inclusion.

```

Finally, the sub-algorithm corresponding to the transformation at ontological level that would merge attributes and relations for equivalent concepts from participant ontologies into the OM can be expressed as follows.

```

Ontological_transformation sub-algorithm
{Let Oi(t) be each mereological child of OM(t) until instant t; Let
N be the total number of mereological children of OM(t)};
For i=1 to N
  For each concept ct belonging to Oi(t) do
    If there is any concept c't belonging to OM(t) such that
      Equivalent_concepts(ct,c't) or
      ct and c't have same name
    Then merge attributes and relationships for ct and c't)
    Else link ct to its parents in OM(t)
  End-for
End-Ontological-transformation.

```



Fig. 1. Display of available options for a remote agent – an expert user contributing to the construction of the OM

3.2 Implementation of the Framework

The framework under study has been implemented, tested and validated, as can be seen in [7], the result been the *Ontoint* tool. Two snapshots of this system corresponding to a working session for an expert user can be contemplated in figures 1 and 2.

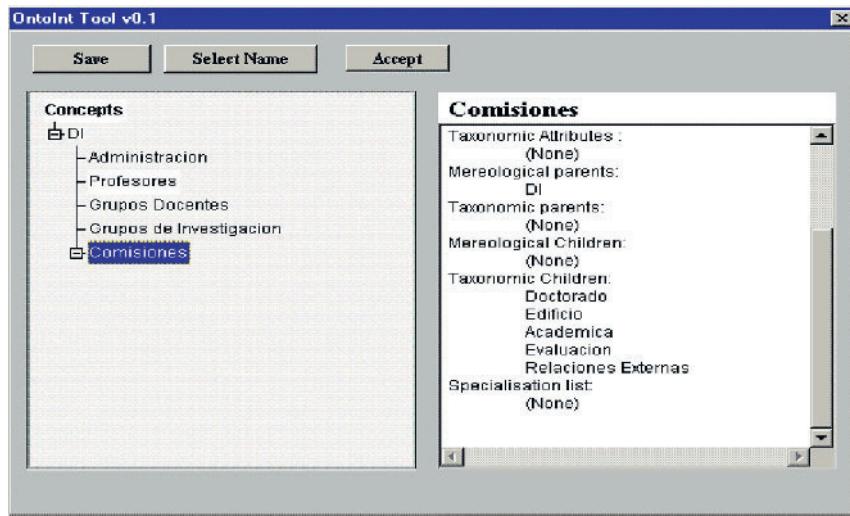


Fig. 2. Visualization of a personalized view of the OM for a certain user

4 Constructing Natural Knowledge: Consensus Versus Personalization

As it has been mentioned before, the framework under study includes a mechanism for avoiding redundancy and inconsistency while maintaining a single ‘macro ontology’ that will keep track of all personalized terminologies (at conceptual level) but that will in the long term maintain a single, yet cooperatively constructed common

OM. The definition proposed for conceptual inconsistency is possibly too restrictive and was originally justified for all the integration process could happen in a totally automatic mode.

The need for consensus can be understood and even justified in this context, given the organizational nature and definitional aspects of the memory. After analyzing the algorithms and the complete framework presented in [7], the construction of the OM from the ontological point of view seems to be rather restrictive, as individual potentially contributing ontologies holding less knowledge – in terms of computable total number of concepts, attributes and relational links – can be left out of the process. The reason for adopting decision criteria based upon quantitative aspects was the requisite of the integration mechanism to be performed in a completely automatic way. The question of the quality, that is, the value of knowledge is not yet resolved, and would possibly require a completely different approach.

On the other hand, the framework under analysis supports personalized access to organizational knowledge, that is, distributed individual agents can solicit a view of the OM constructed so far that will be visualized instantiated with his/her own terminology - including other knowledge items (concepts, attributes or relations) originally created by other users. Individual agents can possess different ontologies, but regarding the OM ontology, management of versions not contemplated. In this sense, the need for (automatic) consensus seems to be playing an antagonist role with regards to personalization, for it seems to be not feasible having real different visions of the knowledge.

5 Towards Semantic Organizational Memories

The limitations found after the analysis of the previous framework had led the author to consider further and substantial extension of the ontological environment originally proposed. Regarding the general interoperability and extensibility of the framework, several restrictions can be found. Firstly, the language for representing ontologies within the OM framework is not a standard, which would oblige any participating agent (human or digital) to know and be able to express ontologies using this particular representation. Secondly, another fundamental restriction is that it is not possible either to enable any automatic/embedded reasoning (intra and extra-ontological) that could be to some extend supported by the ontological representation. Besides, the definition for conceptual inconsistency proposed is possibly too restrictive and was originally justified for all the integration process to happen in a totally automatic mode. It would be possible although those two portions of the same knowledge (corresponding to different expert users) could be found to be inconsistent by the experts under question, that not being detected by the current framework. Another limitation of this approach is that attributes are term-based and more expressiveness would be desirable.

The above reasons would strongly suggest the consideration of other means of representing ontologies. In this sense, semantic web technologies¹ seem to be clearly the

¹ Semantic web technologies refer to new technological standard derived from the Semantic Web Project. URL: <<http://semanticweb.org>>

most promising way to advance in that direction. Amongst this family of knowledge technologies available, OWL, together with its predecessors OIL and DAML+OIL seem to be the trend supported by the latest international effort for unifying ontology representation languages [8]. With respect to the ontological environment under study, it seems possible and viable the adaptation and subsequent translation of the integration mechanism, even our tool in Java to handle integration and personalization with OWL ontologies. Efforts are already made in that direction by the author's research group.

References

1. Campoy-Gómez, L., Martínez-Béjar, R., and Martín-Rubio, F.: A knowledge-based system prototype for environmental engineering. *International Journal of Knowledge-Based Intelligent Engineering Systems*. Vol. 3:4 (1997) 254-261
2. Fernández-Breis, J.T., Martínez-Béjar, R., Campoy-Gómez, L., and Martín-Rubio, F.: A cooperative approach to corporate memory modeling. In: European Conference on Artificial Intelligence, ECAI'2002 Workshop on Knowledge Management and Organizational Memories, Lyon, France, July 21-26 (2002)
3. Benjamins, V.R., Fensel, D. and Gomez-Perez, A.: Knowledge Management through Ontologies. In: Proceedings of the 2nd International Conference of Practical Aspects of Knowledge Management (1998)
4. Milton, N., Shadbolt, N., Cottam, H. and Hammersley, M: Towards a Knowledge Technology for Knowledge Management. *International Journal of Human Computer Studies*. Vol. 51 (1999) 615-641.
5. Reimer, U., Brockhausen, P., Lau, T. and Reich, J.R.: The SwissLife Case Studies. In: Davis, J., Fensel, D. and van Harmelen (Eds): *Towards the Semantic Web: Ontology-driven Knowledge Management*, John Wiley & Sons (2003) 197-217
6. van Heist, G., van der Spek, R. and Kruizinga, E.: Organizing Corporate Memories. In: Gaines, B. and Musen, M. (Eds): *Proceedings of Knowledge Acquisition Workshop*, Banff, Canada, November. Vol. 42 (1996) 1-17
7. Campoy-Gómez, L.: A Formal Approach for the integration of reusable and shareable knowledge components in corporate memories management (In Spanish), PhD Thesis, ISBN-84-8371-381-0, University of Murcia, Spain (2003)
8. Fensel, D., van Harmelen, F. and Horrocks, I.: OIL and DAML+OIL: Ontologies for the Semantic Web. In: Davis, J., Fensel, D. and van Harmelen (Eds): *Towards the Semantic Web: Ontology-driven Knowledge Management*, John Wiley & Sons (2003) 197-217

Improving the Naming Process for Web Site Reverse Engineering

Sélima Besbes Essanaa and Nadira Lammari

292, rue Saint Martin 75141 Paris Cedex 03, France
besbes_s@Auditeur.Cnam.fr, lammari@cnam.fr

Abstract. We have recently defined RetroWeb, an approach to reverse engineer the informative content of semi-structured web sites. This approach provides a description of the web site informative content at physical, logical and conceptual levels. At each level a meta-model is instantiated using a set of reverse engineering rules. This paper focuses on the naming process used to instantiate the meta-models. We introduce an algorithm that will improve the labeling itself by reducing the number of objects to name. This algorithm is based on the analysis of the dependencies describing the inclusion, exclusion and equality between sets of objects.

1 Introduction

This research work describes a naming process used in RetroWeb [1], an approach for web site informative content reverse engineering. RetroWeb gives a description of the informative content of the site at various abstraction levels: physical, logical and conceptual levels. Besides the EER conceptual model, two meta-models are proposed to describe, at these different abstraction levels, the semi-structured data coded on the HTML pages. The first one represents the Web site through its physical views. The second one describes the Web site through its logical views. Mapping rules are proposed for the translation of physical views into logical ones and then into EER conceptual views.

This approach concerns sites that suffer the effects of rapid and unstructured construction and / or evolution processes. The evolution of such Web site has been addressed from various ways. Some research works take an interest in the evolution of the presentation, others in the restructuring of the HTML code and others in the definition of approaches that aim to obtain abstract representation of Web sites. In this last category, we find research works on data extraction from HTML code. In these works a wrapper is manually, semi-automatically or automatically deduced. For manual wrappers, the naming of the extracted data is mechanical since labels are in the body of the wrapper itself. Since semi-automatic wrappers are based on the training techniques from manually labeled examples, the semantic of the extracted data is defined by the user according to the labels that have been attributed to the fields to be extracted from the training examples. Automatic wrappers are those deduced without soliciting the user [2, 3, 4, 5]. A naming step is, so, needed to reduce human intervention. In [5], for example, the tool developed to annotate extracted data is based on heuristics that use spatial relationships among a page to associate a piece of data to an

invariant string, which co-occurs with it [6]. The DeLa system [2] proposes heuristics to label the extracted data by matching form element labels to the extracted data.

The naming in RetroWeb uses a process that will improve the labeling itself by reducing the number of objects to name. This process uses the normalization algorithm defined in [7]. This latter is based on the analysis of the dependencies describing the inclusion, exclusion and equality between sets of objects.

The rest of the paper is organized as follows. Section 2 is dedicated to a brief presentation of RetroWeb through an example. Section 3 describes the RetroWeb naming process and gives an overview of the normalization algorithm. Section 4 concludes the paper and presents some perspectives.

2 The RetroWeb Approach

RetroWeb reverse engineers the informative content of semi-structured and undocumented Web sites. Through a simple example of a complete execution of RetroWeb, we will attempt, in the following, to describe this approach. More details about it can be found in [1]. The example concerns a Web page that displays, for each volume of an academic journal, its authors. Fig. 1 presents the page and its corresponding HTML code.

```

<HTML>
...
<BODY>
...
<LI>
<A HREF="http://books/index.html"> Volume N° 19 (3) </A>
<B>Competitive Strategy, Economics, and the Internet</B>
<A HREF="http://books?Chiraru+719"> Chiraru </A> <P>Alina M.
<A HREF="http://books?Kauffman+158"> Kauffman</A> <P>Robert J.

<LI>
<A HREF="http://books/index.html"> Volume N° 19 (2) </A>
<B>Enterprise Resource Planning</B>
<A HREF="http://books?Ragowsky+701"> Ragowsky</A> <P>Arik

<LI>
...
</BODY>
</HTML>

```

Fig. 1. An academic journal publications Web site page and its corresponding HTML code

RetroWeb encompasses three steps: extraction, conceptualization and integration. The extraction step aims to the retrieval of the semi-structured data coded on each HTML page of the site and to describe them through physical views (one physical view per HTML page). It is performed into three phases: pre-processing, extraction and naming phases. The pre-processing phase takes as an entry HTML pages, corrects them, proceed to some cleaning, executes some transformations and then returns, for each page, a coded sequence describing the structure of the page. In this sequence, the structure tags are codified and all textual data that are not identified as tags are replaced by a token "Text". The second phase deduces pattern expressions that will be used by the wrapper to extract data from pages. The last phase assigns names or labels to variables of the physical views. Indeed, the precedent phase being automatic, no names have been affected to the detected variables.

The result of the execution of this step on the Web page of Fig. 1 is the physical view of Fig. 2a. A terminal node of the tree represents the set of values taken by the associated simple variable in the various instances or blocks as they are displayed in the HTML code. Each simple variable is encapsulated in a composed-type variable. This latter can be also composed by multi-valued type variables.

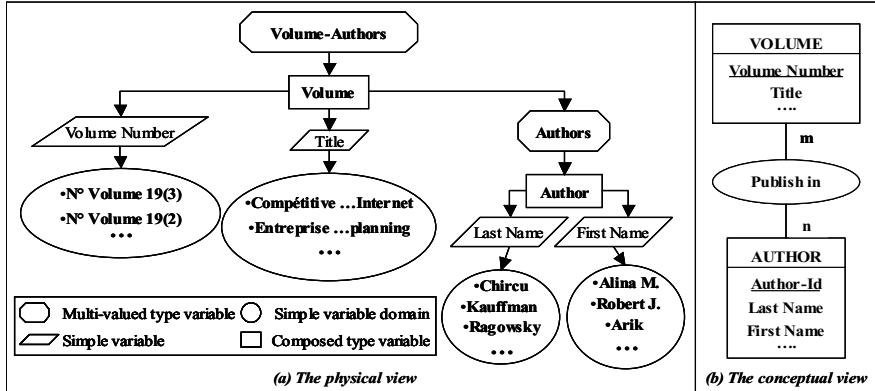


Fig. 2. The physical and the conceptual views of the Web page of Fig. 1

The conceptualization step aims to produce the EER schemas associated with the physical views. The result of the execution of this step on the physical view of Fig. 2a is the conceptual schema of Fig. 2b. In order to reach this result, the conceptualization step translates first the physical views into logical ones, constructs for each logical view its corresponding EER conceptual schema and then, affects significant labels to their entity-types and relationship-types. The different schema transformations are performed thanks to reverse engineering rules.

The integration step merges the portions of EER schemas into a global one in order to give a global conceptual description of the whole web site informative content. This step is based on integration techniques well known in the information systems context.

3 The Naming Process

The naming is performed on physical views and on entity-types of conceptual views. It is decomposed into two phases. The first phase is a concept classes definition phase. It aims to reduce the number of concepts to name. It defines classes of concepts that may be assigned the same label. The second phase assigns to any concept, not yet labeled, a name and gives this name to its family (i. e. to all concepts sharing the same class). It uses, to that end, some heuristics.

3.1 The Algorithm for Defining Concept Classes

Two concepts (variables or entities) that appear in two different views can either have or not the same semantics or two different semantics. In the case where they have equal definition domains, there is every chance that they represent the same semantics. In this case, they can share the same label. In the other hand, if their definition domains are disjoint, there is every chance that they represent different semantics. Finally, if the definition domain of a concept C1 is included into the definition domain of a concept C2, then we can assume that the semantics of C2 covers the semantics of C1. On other words, in the case where the naming phase don't succeed in finding a label for C1, we can assign it with the label of C2. To determine which concepts can take the same name, we propose to apply the normalization algorithm proposed in [7]. This algorithm builds an IS_A hierarchy starting from a set of concepts.

This algorithm is based on a set of constraints describing the inclusion, exclusion and equality of concept's definition domains. In this paper context, the definition domain of a concept is: (i) a value domain if the concept is a physical view simple type variable or (ii) a set of variable constructing the concept if the latter is a composed type or a multi-value type variable of a physical view or (iii) a set of properties describing the concept if this one is an entity-type. The defined constraints¹ are three kinds: mutual, exclusive and conditioned ones. A *mutual constraint* defined between two concepts x and y, denoted $x \leftrightarrow y$, describes the equality of their definition domains. An *exclusive constraint* defined between two concepts x and y, denoted $x \nleftrightarrow y$, describes the disjunction of the two definition domains. Finally, a *conditional constraint* defined between two concepts x and y, denoted $x \rightarrow y$, captures the fact that the definition domain of x is included into the definition domain of y.

The main idea of the normalization algorithm is to group in the same class all concepts sharing the same definition domain and to construct an IS_A hierarchy according to the inclusion links between classes. In our web site reverse engineering context, the normalization mechanism can be applied either for entity-types or for variables.

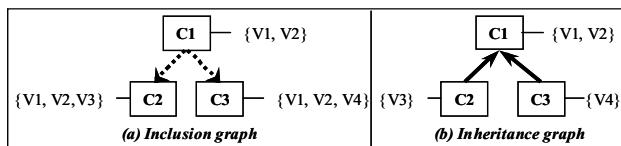


Fig. 3. The result of the application of the normalization of the set {V1, V2, V3, V4}

For instance, let us consider four simple variables V1, V2, V3 and V4 having as value domain respectively DV1, DV2, DV3 and DV4. Let us also consider the following constraints: $V1 \leftrightarrow V2$, $V3 \rightarrow V1$, $V4 \rightarrow V1$, $V3 \nleftrightarrow V4$ which respectively means that $DV1 = DV2$, $DV3 \subset DV1$, $DV4 \subset DV1$ and $DV1 \cap DV2 = \emptyset$. According to these constraints, the four variables constitute three classes C1, C2 and C3 sharing the same value domain (see Fig 3a). The organization of these three classes according to the inclusion links between them leads to the IS_A hierarchy of Fig. 3b.

¹ Their formal definitions and generalization to a set of concepts can be found in [8].

This algorithm encompasses three phases. First, it deduces, from the set of concepts, valid classes. Second, it organizes these valid classes into an inclusion graph and then, by inverting the direction of the graph inclusion arrows, it deduces the IS_A hierarchy. A valid class is a subset of the concepts set that is compatible with the domain definition constraints².

The concern of the concepts definition phase at the extraction step is to apply the normalization algorithm three times: (i) to the set of simple variables of all the extracted physical views, (ii) to the set of composed type variables of all extracted physical views after replacing the simple variables by their associated classes and (iii) finally to the set of multi-valued type variables of all extracted physical views after replacing the treated variables by their associated classes. We have also to apply this algorithm to the set of entity-types of unnamed EER views.

Let us remark that some intersection cases between concept definition domains can be translated as equality, inclusion or disjunction of domains. For instance, we can assume that two concepts are related by a mutual constraint if their value domains are nearly equal, i. e. their intersection is important and their differences are insignificant. To take into account all the situations, a threshold must be defined, by the reverse engineer, at the beginning of the process naming in order to define the bigness or smallness of intersections and differences.

3.2 Assigning Names to Concepts

As said in the previous paragraph, the naming concerns classes of concepts. In other words, when a concept is named, by the reverse engineer or using a heuristic, the assigned label is also assigned to each concept of the same class. At the present time, the naming of entities, composed type and multi valued type variables is totally manual in this sub-phase. We wish to improve this process in future work by using domain ontologies. On the other hand, we introduce some automation during the naming of simple variables. We propose some heuristics that will assign automatically labels to simple variables. Their objective is to identify text strings (potential labels) that can represent the meaning of the extracted data. For example, textual data can contain invariant strings in all occurrences of the same token “Text”. This invariant string can be a candidate label for the simple variable representing the data. Moreover, for the dynamic pages (generated from a database), the comparison of the extracted domains with the database domains leads to choose the corresponding column title as a label for the associated simple variable. We can also exploit the syntax of data contained in a value domain in order to determine their semantics. For example, if a value domain contains the symbol “@”, we can deduce that the corresponding simple variable represents an electronic addresse.

Since several candidate labels can be proposed by different heuristics, we propose to assign weights to heuristics. The most appropriate candidate label will be the one that results from the heuristic with the biggest weight.

² Its formal definition can be found in [7]

4 Conclusion

Through RetroWeb, we wish to recover the informative content of a whole web site. Thanks to the proposed meta-models, RetroWeb supplies, at each abstraction level (physical, logical and conceptual levels), a clear and semi-formal description of the web site informative content. The mapping between the meta-models is performed using reverse engineering rules.

The originality of this research work is the application of an algorithm, based on mutual, exclusion and conditioned constraints, that allows to recover concepts, organized into an IS_A hierarchy, from a flat set of data dispatched through all the pages of the web site.

We have then initiate a research work concerning the naming of the concepts. At the present time, we have proposed a set of heuristics that allows finding, in some cases, names. The enrichment of the set of heuristics is in progress. We expect to study the applicability of learning approaches for the naming [9], in the context of web site reverse engineering. We believe that the use of ontologies, such as [10], will largely contribute in the naming process.

References

1. Essanaa, S., Lammari, N.: RetroWeb: Une approche de rétro-conception de contenu informatif de sites Web. Proc. of the 8th Maghrebian Conf. on Software Engineering and Artificial Intelligence, sousse may (2004)
2. Wang, J., Lochovsky, F.: Data Extraction and Label Assignment for Web Databases. Proc. 12th Int. Conf. on World Wide Web, Hungary (2003) 187–196
3. Buttler, D., Liu, L., Pu, C.: A fully Automated Object Extraction System for the World Wide Web. Proc. Int. Conf. on Distributed Computing Systems, (2001) 361–370
4. Chang, C. H., Lui, S. C.: IEPAD: Information Extraction Based on Pattern Discovery. Proc. 10th Int. Conf. on World Wide Web, Hong Kong may (2001) 681–688
5. Crescenzi, V., Mecca, G., Merialdo, P.: ROADRUNNER: Towards Automatic Data Extraction from Large Web Sites. Proc. 27th Int. Conf. on Very Large Data Base, (2001) 109–118
6. Arlotta, L., Crescenzi, V., Mecca, G., Merialdo, P.: Automatic Annotation of Data Extracted from Large Web Sites. Proc. 6th Int. Workshop on the Web and Databases, San Diego (2003) 7–12
7. Lammari, N., Laleau, R., Jouve, M.: Multiple Viewpoints of Is_A Inheritance Hierarchies through Normalization and Denormalization Mechanisms. Proc. Int. Conf. on Object-Oriented Information systems, Springer-Verlag, Paris september (1998) 9-11
8. Lammari, N.: Réorganisation des Hiérarchies d'Héritages dans un Schéma Conceptuel Objet. Phd thesis, Conservatoire National des Arts et Métiers, october 24 (1996)
9. Meziane, F., Kasiran, M. K.: Extracting Unstructured Information from the WWW to support Merchant Existence in eCommerce. 8th Int. Conf. on Applications of Natural Language to Information Systems, (2003)
10. Lenat, D. B., Millar, G. A., Yokoi, T.: CYC, WordNet, and EDR: Critiques and Responses". In CACM 38 (11), (1995) 45-48

On Embedding Machine-Processable Semantics into Documents

Krishnaprasad Thirunarayan

Department of Computer Science and Engineering
Wright State University, Dayton, OH 45435, USA.
`tkprasad@cs.wright.edu`
<http://www.cs.wright.edu/~\tkprasad>

Abstract. Most Web and legacy paper-based documents are available in human comprehensible text form, not readily accessible to or understood by computer programs. Here we investigate an approach to amalgamate XML technology with programming languages for representational purposes. Specifically, we propose a modular technique to embed machine-processable semantics into a text document with tabular data via annotations, and evaluate it vis a vis document querying, manipulation, and integration. The ultimate aim is to be able to author and extract, human-readable and machine-comprehensible parts of a document “hand in hand”, and keep them “side by side”.

1 Introduction

The World Wide Web currently contains about 16 million web sites hosting more than 3 billion pages, which are accessed by over 600 million users internationally. Most of the information available on the web, including that obtained from legacy paper-based documents, is in human comprehensible text form, not readily accessible to or understood by computer programs. (Quoting from SHOE FAQ, “Web is not only written in a human-readable language (usually English) but in a human-vision-oriented layout (HTML with tables, frames, etc.) and with human-only-readable graphics”. [6]) The enormity and the machine incomprehensibility of the available information has made it very difficult to accurately search, present, summarize, and maintain it for a variety of users [1]. Semantic Web initiative attempts to enrich the available information with machine-processable semantics, enabling both computers and humans to complement each other cooperatively [3,7]. Automated (web) services enabled by the semantic web technology promises to improve assimilation of web content.

XML (eXtensible Markup Language) is a meta-language for designing customized markup languages for different document types. Conceptually, an XML document consists of tagged text (that is, markup is inter-twined with text) in which the tag makes explicit the category and the properties of the enclosed text using attribute-value pairs. In general, existing XML technology can be applied to formalize, transform, and query text documents.

However, the XML technology developed so far cannot be readily used to formalize/render heterogeneous documents (e.g., MS Word document containing text, images, and complex data structures (such as numeric tables)) in a form that is suitable for semantic web applications. Specifically, the current approaches to document representation and authoring do not directly address the issue of preserving or abstracting the superficial structure of data (e.g., rectangular grid presentation format for tables) that is suitable for human consumption and traceability, while simultaneously making explicit semantics of data for machine processing. Furthermore, in order to tap into existing legacy documents, it is necessary to develop “modular” and “linear” techniques to augment documents with machine processable semantics.

In this paper, XML-based programming languages are evaluated to determine how well they can serve as a substrate for embedding machine-processable semantics into text documents containing complex data. In Section 2, we motivate an XML-based programming and representation language. In Section 3, we consider an approach to formalizing tabular data embedded in text document, *in-place*, via a concrete example. In Section 4, we evaluate the proposed approach by presenting the pros and the cons, critiquing it in the context of real-world documents such as Materials and Process Specifications. In Section 5, we conclude with suggestions for long-term research.

2 XML-Based Programming and Representation Language

An XML-Schema can be used to specify a standard syntax for information exchange. Overlaying domain-specific XML tags on a text document enables *abstraction*, *formalization*, and *in-place embedding* of machine-processable semantics. However, this approach still yields *static declarative* data, not conveniently handled by extant programming languages. The “impedance mismatch” has been dealt with by providing APIs to mediate conversions to and from XML and native data structures. Furthermore, in a number of distributed applications, an XML document can encode behaviors, whose dynamic semantics can only be uncovered by an embedded interpreter in the host language. To remedy the need for multiple languages and to assist in building web services, XML-based programming languages such as Water [5], XPL [8], etc have been developed.

In order to embed machine-processable semantics into a document, we consider XML-based language for knowledge representation and for encoding behavior consisting of: (i) data in the form of text with clearly marked semantic annotations, and (ii) behavior/program in the form of definitions for functions, classes, etc. Ordinary text corresponds to human sensible part, while annotation (with their definitions) corresponds to machine processable part. Definitions encapsulate behavior and can be used to provide and carry out machine processable semantics of data. For instance, the text data “**Delhi is the capital of India.**” can be formalized to different levels of detail in terms of domain-specific vocabulary as follows:

```

<capital_of India "New Delhi">
    <city name="New Delhi"> Delhi </> is the capital of
    <country name=India> India </>.
</>

```

(Note that Concise XML Syntax does not require *end tag names* to be explicit.)

Formalization requires recognizing and delimiting text that corresponds to a relevant concept, and then mapping the delimited text into a standard form that captures its intent. Each resulting annotation consists of an associated XML-element that reflects the semantic category to which the corresponding text fragment belongs, and the XML-attributes are bound to the relevant semantic values. Furthermore, *the annotated data can be interpreted by viewing it as a function/procedure call, and defining the XML-element as a function/procedure*. The correspondence between formal parameters and actual arguments can be positional or name-based. Additionally, the definition can make explicit the type of each formal argument, or provide additional integrity constraints to be satisfied by the actual arguments, or in general, map the semantic values. The same annotated data can be interpreted differently by programming-in different behaviors for the XML-element.

To summarize, the idea of semantic markup of text is analogous to overlaying the abstract syntax (with attributes) on the free-form text such that the resulting XML document can be flexibly processed by associating different collections of behaviors with XML-elements additively.

3 Formalizing Tabular Data In-Place

In relational databases, tables contain schema information as row/column headings and data as rows/columns of entries. In the realm of heterogeneous documents, for example, a table (built out of table primitives or just hand formatted) may be present within an MS Word document, that needs to be isolated, abstracted, and saved as plain text and formalized, before any semantic processing can begin. To motivate and illustrate an XML-based approach to providing semantics to complex data found in text, consider representation of such tables. Assume that the table contains both the headings and the data entries. The precise relationships among the various values in a row/column are tacit in the headings, and are normally obvious to the domain expert (human reader). However, this semantics needs to be made explicit to do any machine processing, including querying, integration, and formal reasoning. If, on the other hand, only semantics rich translation is stored, it may not always be conducive to human comprehension. Thus, it is useful to have a representation language that can serve both the goals. That is, the representation language should have the provision to more or less preserve the grid layout of a table so that changes to the original table can be easily incorporated in text, but, at the same time, describe the semantics of each row/column in a way that is flexible, and applicable to all rows/columns for further machine manipulation.

An XML-based programming language seems to provide a balanced way to achieve and integrate “best” for both the worlds:

- to encode data and to make explicit the semantics in a modular fashion, and
- to effectively use this information for formal manipulation.

For example, the following common table form (found in materials and process specifications) can be saved as text

Thickness (mm)	Tensile Strength (ksi)	Yield Strength @0.2% offset (ksi)
0.5 and under	165	155
0.50 - 1.00	160	150
1.00 - 1.50	155	145
...

and subsequently annotated as shown below:

```
<table type=Strength>
<parameter name="Yield Offset" value="0.2%"/>
<rowHeadings "Thickness" "Tensile Strength" "Yield Strength"/>
<rowData 0 0.50 165 155 />
<rowData 0.50 1.00 160 150 />
<rowData 1.00 1.50 155 145 /> ...
</table>
```

This, when augmented with `Strength` table definition, should yield a `Strength` table appropriately initialized and exhibiting a prescribed semantics.

In the absence of an implemented XML-based programming language at this time, for concreteness, we will use Water-like syntax to formalize the semantics of a smaller table below. (Note that Water syntax does not permit embedding code in free-form text and there are no global variables. So, to test the following code in Water, the document text must be deleted [5].)

```
Thickness (mm)      Tensile Strength (ksi)      Yield Strength (ksi)
table.<setHeading thickness strength.tensile strength.yield/>
0.50 and under    165                      155
table.<addRow 0 0.50 165 155 />
0.50 - 1.00       160                      150
table.<addRow 0.50 1.00 160 150 />
1.00 - 1.50       155                      145
table.<addRow 1.00 1.50 155 145 /> ...
```

Each row is annotated with a tag which becomes a method invocation on a table object. If the rows require dissimilar treatment, different tags can be used.

```
<defclass thickness value=required=number units="mm"/>
<defclass thicknessRange max=required=number min=optional=0 units="mm"/>
<defclass strength value=required=number units="ksi">
    <defclass tensile/>      <defclass yield offset="0.2%"/>
</defclass>
<defclass table rows=required=vector heading=optional=vector>
```

```

<defmethod setHeading t=required ts=required ys=required>
  <set heading=<vector t ts ys/>/>
</>
<defmethod addRow smin smax ts ys>
  <set rows=table.rows.<insert <vector smin smax ts ys/>/>/>
</>
<defmethod computeYieldStrength> ...
</>
<defmethod computeTensileStrength>
  <set temp=fluid.Thickness/>      <set i=0/>
  <do>
    <until <and temp.<less table.rows.<get i/>.1/>
           temp.<more_or_equal table.rows.<get i/>.0/> /> >
      table.rows.<get i/>.2
    </until>
    <set i=i.<plus 1/>/>
  </do>
</> </>
fluid.<set Thickness=0.60>
<try   <set TensileStrength=table.<computeTensileStrength/>/>
      YieldStrength
    > "TABLE: out of range error occurred"
</try>
</set> ...

```

The annotated data can be processed using constructs that exploit uniformity. For instance, a looping construct can abbreviate dealing with rows, or a primitive function can be used to split a single (suitably delimited) string into component values, *preserving linear relationship between annotated data and the original text*. Ideally, tabular data in each document is annotated, while factoring out annotation definitions separately as “background knowledge”.

4 Evaluation

Microsoft’s Smart Tags technology enables recognition of strings from a controlled vocabulary and associate every occurrence of such strings in a document created using MS Office 2003 with a list of predefined actions [4]. In comparison, the technique discussed here advocates tagging an existing text document and programmatically describing actions associated with tags using XML-syntax. This approach can formalize relationships described in text, and enable ultimately to author and extract, human-readable and machine-comprehensible parts of a document “hand in hand”, and keep them “side by side”. As to the generalization for tabular data is concerned, it is viable if there are fewer distinct table forms (semantics) compared to the number of concrete data tables. Given that the semantics of a rectangular grid of numbers is implicit in the text, this approach provides a means to make the semantics explicit. However, it uses a functional style requiring detailed description of the answer extraction process. To compare this to querying of declarative table data in logic programming style, along the lines of SHOE [6], consider the following encoding in Prolog.

```

strengthTableRow( 0,    0.50,   165, 155).
strengthTableRow(0.50, 1.00,   160, 150).
strengthTableRow(1.00, 1.50,   155, 145).      ...
strengthTable(Thickness, TensileStrength, YieldStrength) :-
    strengthTableRow(L, U, TensileStrength, YieldStrength),
    L <= Thickness, U > Thickness.
thicknessToTensileStrength(Thickness, TensileStrength) :-
    strengthTable(Thickness, TensileStrength, _).
thicknessToYieldStrength(Thickness, YieldStrength) :-
    strengthTable(Thickness, _, YieldStrength).
?- thicknessToYieldStrength(0.6,YS).

```

The “regular” tables considered so far exemplify relatively tractable scenario for complex data. In practice, these tables can serve as target encodings of more complex data. Overall, the proposed technique is still not convenient for document integration, which requires normalization.

5 Conclusions

This paper made a case for developing an XML-based Programming and Representation language for authoring and extracting extant Web and legacy documents. In order to appreciate the hurdles to be overcome, we considered the formalization of heterogenous documents. The approach discussed is modular in that the annotation of the original document and the codification of the semantics of annotations can be separated. The approach enables adequate formalization of document, that can support traceability and querying. However, it is still not convenient for document integration, or automatic tagging of tables.

For the long-term success of the Semantic Web initiative, it is important to understand principles and develop techniques/tools for authoring and formalizing documents in a form that is both human comprehensible and machine processable. For tractability and concreteness, it may be beneficial to scope the domain of discourse.

References

1. Davies J., Fensel D., and van Harmelen F. (Eds.): *Towards the Semantic Web: Ontology-Driven Knowledge Management*, John Wiley and Sons, Inc., 2003.
2. Fensel D.: Semantic Web enabled Web Services, Invited Talk at: In *7th Intl. Conference on Applications of Natural Language to Information Systems*, June 2002.
3. Fensel, D., et al (Eds.): *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, The MIT Press, 2003.
4. Kunicki Chris: What's New with Smart Tags in Office 2003, MSDN Library Article, Microsoft Corporation, January 2003.
5. Plusch M.: Water : Simplified Web Services and XML Programming, Wiley Publishing, 2003. (<http://www.waterlanguage.org/>, Retrieved 4/20/2004.)
6. <http://www.cs.umd.edu/projects/plus/SHOE/index.html>, Retrieved 4/20/2004.
7. <http://www.semanticweb.org/>, Retrieved 4/20/2004.
8. <http://www.vbxml.com/xpl/>, Retrieved 4/20/2004.

Using IR Techniques to Improve Automated Text Classification

Teresa Gonçalves and Paulo Quaresma

Departamento de Informática, Universidade de Évora,
7000 Évora, Portugal
`{tcg | pq}@di.uevora.pt`

Abstract. This paper performs a study on the pre-processing phase of the automated text classification problem. We use the linear Support Vector Machine paradigm applied to datasets written in the English and the European Portuguese languages – the Reuters and the Portuguese Attorney General’s Office datasets, respectively.

The study can be seen as a search, for the best document representation, in three different axes: the feature reduction (using linguistic information), the feature selection (using word frequencies) and the term weighting (using information retrieval measures).

1 Introduction

In the last years text classification is gaining popularity due to the increased availability of documents in digital form and the following need to access them in flexible ways. This problem is well known in the Information Retrieval community and the use of Machine Learning techniques is opening many important and interesting research problems.

Research aimed at the application of Machine Learning methods to text classification has been conducted among others by Apté et al. (rule-based induction methods) [1], Mladenić and Grobelnik (naïve Bayes) [7], Nigam et al. (EM and naïve Bayes) [6] and Joachims (SVM – support vector machines) [5].

In Joachims’s work, documents are represented as bag-of-words [9] (without word order information) and the results are evaluated using information retrieval measures, such as the *precision recall break-even point* (PRBP).

In this paper, we follow his approach, aiming to determine if linguistic information is helpful for achieving good SVM performance. We use two sets of documents written in two different languages – the European Portuguese (the PAGOD dataset [8]) and the English one (the Reuters dataset).

The work can be seen as a search in three different axes: the feature reduction (using linguistic information), the feature selection (using word frequencies) and the term weighting (using information retrieval measures) axes.

On previous work, we evaluated SVM performance compared with other Machine Learning algorithms [2] and performed a preliminary study on the impact of using linguistic information to reduce the number of features [3]. In this paper, we extend that work using IR techniques to weight and normalise features.

In Section 2 a brief description of the Support Vector Machines theory is presented, while in Section 3 our classification problem and datasets are characterised. Our experiments are described in Section 4 and the results are presented in Section 5. Finally, some conclusions and future work are pointed out in Section 6.

2 Support Vector Machines

Support Vector Machines (SVM) belong to the group of kernel learning algorithms. These algorithms come from the area of statistical learning theory and are based on the structural risk minimisation principle [11].

SVM are supervised binary linear classifiers and, as such, they fail to present a solution when the boundary between the two classes is not linear. In this situation the approach followed is to project the input space X into a new feature space F and try to define a linear separation between the two classes in F . In this way, SVM classifiers can be obtained using algorithms that find the solution of a high dimensional quadratic problem.

In the scope of this work only linear kernels, the functions that transform the input feature space, are used. More detailed information can be obtained in several specialised books, such as [10].

3 Domain Description

The text classification problem at hand (both, the Reuters and the PAGOD datasets), can be characterised as a multi-label one, i.e. documents can be classified into multiple concepts/topics. The typical approach to solve it, is to divide into a set of binary problems, where each concept is considered independently, reducing the initial problem to several binary classification ones.

An important open problem is the representation of the documents. In this work, as already mentioned, we will use the standard vector representation, where each document is represented as a bag-of-words. We discarded all words containing digits and retained words' frequencies.

3.1 The Reuters Dataset

The Reuters-21578 dataset was compiled by David Lewis and originally collected by the Carnegie group from the Reuters newswire in 1987. We used the *ModApte* split, that led to a corpus of 9603 training and 3299 testing documents.

On all 12902 documents, we found 31715 distinct words; per document, we obtained averages of 126 words, of which 70 were distinct.

3.2 The PAGOD Dataset

This dataset has 8151 documents and represent the decisions of the Portuguese Attorney General's Office since 1940. It is written in the European Portuguese

language, and delivers 96 MBytes of characters. All documents were manually classified by juridical experts into a set of classes belonging to a taxonomy of legal concepts with around 6000 terms.

From all potential categories, a preliminary evaluation showed that only about 3000 terms were. We found 68886 distinct words and, per document, we obtained averages of 1339 words, of which 306 were distinct.

4 Experiments

We chose the top five concepts and applied the SVM learning algorithm using a linear kernel. For each dataset we performed three classes of experiments: a feature reduction one (using linguistic information), a rudimentary kind of feature selection and some term weighting techniques (from the IR field). For each experiment we analysed the precision, recall and F_1 measures [9].

We generated a linear SVM for each possible combination of the experiments' classes, using the WEKA package [12] from Waikato University, with default parameters. For the Reuters dataset we used the training and test sets, while for the PAGOD dataset we performed a 10-fold cross validation procedure.

4.1 Feature Reduction

On trying to reduce the number of features we made three different experiments: in rdt_1 we used no linguistic information, in rdt_2 we removed a list of considered non-relevant words (such as articles, pronouns, adverbs and prepositions) and in rdt_3 we removed the same non-relevant words and transformed each remaining word onto its stem (its lemma for the Portuguese language).

In the Reuters dataset we used the FreeWAIS stop-list to remove the non-relevant words and the Porter algorithm to transform each word onto its stem. In the PAGOD dataset, this work was done using a Portuguese lexical database, POLARIS, that provided the lemmatisation of every Portuguese word.

4.2 Feature Selection

Feature selection was done by eliminating the words that appear less than a specific number in the set of all documents: for example, sel_{55} means that all words that appeared less than 55 times in all documents were eliminated. We performed experiences for sel_1 , sel_{50} , sel_{100} , sel_{200} , sel_{400} , sel_{800} and sel_{1600} .

4.3 Term Weighting

Term weighting techniques usually consist of three components: the document, the collection and the normalisation components [9]. For the final feature vector x , the value x_i for word w_i is computed by multiplying the three components.

We tried four different combinations of components: wgt_1 is the *binary representation* with no collection component but normalised to unit length; wgt_2 uses

the raw term frequencies (TF) with no collection component nor normalisation; wgt_3 uses TF with no collection component but normalised to unit length; wgt_4 is the popular $TFIDF$ representation (TF divided by the document frequency, DF , i.e. the number of documents in which w_i occurs at least once) normalised to unit length.

5 Results

For reasons of space we only show, for each dataset, the values obtained for the micro and macro averaging of the F_1 -measure.

Analysing Reuters' results (Table 1), one can say that, for the feature selection axis, the wgt_3 experiment presents the best F_1 values (both maximum and average values). On the feature reduction axis, and taking into account the previous choice, red_3 is the best experiment. Finally, and for the remaining axis, the sel_{100} experiment is the one that presents the best values. These choices are valid for both macro and micro averaging.

On the other hand, for the PAGOD dataset (Table 2) and using the same procedure, wgt_1 and wgt_3 present best results for the feature selection axis and red_2 and red_3 are the best experiments for the feature reduction one. Concerning

Table 1. Micro and macro F_1 for the Reuters dataset.

		micro				macro			
		wgt_1	wgt_2	wgt_3	wgt_4	wgt_1	wgt_2	wgt_3	wgt_4
red_1	sel_1	0.926	0.891	0.930	0.932	0.859	0.792	0.874	0.874
	sel_{50}	0.918	0.905	0.933	0.930	0.855	0.823	0.887	0.886
	sel_{100}	0.919	0.898	0.933	0.928	0.858	0.798	0.882	0.880
	sel_{200}	0.920	0.894	0.929	0.930	0.858	0.792	0.873	0.876
	sel_{400}	0.920	0.888	0.923	0.924	0.858	0.767	0.855	0.859
	sel_{800}	0.897	0.860	0.898	0.901	0.808	0.700	0.800	0.809
	sel_{1600}	0.854	0.809	0.855	0.849	0.726	0.558	0.703	0.690
red_2	sel_1	0.926	0.888	0.931	0.928	0.863	0.790	0.876	0.869
	sel_{50}	0.920	0.905	0.936	0.929	0.866	0.823	0.888	0.882
	sel_{100}	0.923	0.899	0.937	0.935	0.872	0.806	0.886	0.889
	sel_{200}	0.923	0.897	0.936	0.932	0.866	0.800	0.885	0.877
	sel_{400}	0.924	0.884	0.927	0.926	0.867	0.760	0.865	0.862
	sel_{800}	0.895	0.844	0.893	0.889	0.813	0.680	0.799	0.795
	sel_{1600}	0.841	0.759	0.833	0.832	0.721	0.488	0.622	0.624
red_3	sel_1	0.923	0.889	0.936	0.930	0.861	0.799	0.882	0.874
	sel_{50}	0.920	0.902	0.934	0.931	0.862	0.824	0.882	0.878
	sel_{100}	0.924	0.900	0.937	0.937	0.868	0.813	0.889	0.891
	sel_{200}	0.921	0.898	0.935	0.933	0.866	0.806	0.884	0.878
	sel_{400}	0.921	0.886	0.932	0.928	0.862	0.766	0.873	0.864
	sel_{800}	0.914	0.863	0.913	0.910	0.839	0.708	0.821	0.815
	sel_{1600}	0.844	0.786	0.852	0.845	0.698	0.521	0.689	0.641

Table 2. Micro and macro F_1 for the PAGOD dataset.

		micro				macro			
		wgt ₁	wgt ₂	wgt ₃	wgt ₄	wgt ₁	wgt ₂	wgt ₃	wgt ₄
red ₁	sel ₁	0.759	0.687	0.732	0.722	0.652	0.531	0.631	0.620
	sel ₅₀	0.750	0.694	0.694	0.678	0.651	0.509	0.601	0.587
	sel ₁₀₀	0.747	0.692	0.712	0.700	0.652	0.497	0.615	0.604
	sel ₂₀₀	0.744	0.694	0.731	0.720	0.649	0.502	0.634	0.619
	sel ₄₀₀	0.734	0.688	0.743	0.737	0.644	0.485	0.641	0.629
	sel ₈₀₀	0.730	0.659	0.746	0.740	0.635	0.464	0.638	0.620
	sel ₁₆₀₀	0.745	0.579	0.754	0.744	0.642	0.402	0.632	0.606
red ₂	sel ₁	0.760	0.687	0.757	0.754	0.660	0.533	0.659	0.654
	sel ₅₀	0.750	0.697	0.735	0.737	0.652	0.514	0.640	0.640
	sel ₁₀₀	0.750	0.692	0.740	0.738	0.655	0.500	0.646	0.643
	sel ₂₀₀	0.745	0.691	0.747	0.748	0.656	0.497	0.655	0.655
	sel ₄₀₀	0.740	0.690	0.756	0.756	0.650	0.488	0.661	0.656
	sel ₈₀₀	0.743	0.659	0.754	0.750	0.644	0.467	0.650	0.633
	sel ₁₆₀₀	0.754	0.574	0.763	0.749	0.645	0.399	0.646	0.610
red ₃	sel ₁	0.751	0.673	0.752	0.747	0.652	0.493	0.658	0.653
	sel ₅₀	0.751	0.677	0.746	0.740	0.656	0.480	0.654	0.647
	sel ₁₀₀	0.744	0.672	0.748	0.740	0.650	0.474	0.655	0.643
	sel ₂₀₀	0.742	0.674	0.754	0.749	0.649	0.475	0.661	0.651
	sel ₄₀₀	0.740	0.671	0.761	0.758	0.647	0.473	0.667	0.656
	sel ₈₀₀	0.750	0.631	0.759	0.756	0.652	0.449	0.655	0.637
	sel ₁₆₀₀	0.743	0.560	0.758	0.745	0.630	0.398	0.638	0.603

the feature selection, it is not possible to get a winning experiment. These results are also valid for the micro and macro averaging F_1 values.

6 Conclusions and Future Work

From the previous section and for both datasets, one can reason out that the best term weighting technique is the one that counts term frequencies and normalises it to unit length. From feature reduction results, one can say that linguistic information is useful for getting better performance.

Concerning the feature selection experiments, it is not possible to reach a conclusion valid for both datasets: for the Reuters we have a winning experiment (sel_{100}) while for the PAGOD we have not. This can a characteristic of the written language or of the documents by themselves (for example, on average, the Reuters documents are shorter than the PAGOD ones). Nevertheless, it is possible to say that one can build better classifiers, quicker without loosing performance. Just as an example, and for the PAGOD dataset we are talking on a reduction from almost 6 hours (for sel_1) to 1 hour and half (sel_{400}) for the wgt_3 - rdt_1 experiment.

As future work, we intend to add another axis on our study: the selection of the best features that describe each concept. Instead of word frequencies,

we intend to use other measures, like the Mutual Information from Information Theory.

We also intend to study the impact of the imbalance nature of these datasets on the SVM performance. In fact, there are much more negative examples than positive ones on the binary classifiers and this can be a source of bad results as referred for instance in [4].

Going further on our future work, we intend to address the document representation problem, by trying more powerful representations than the bag-of-words used in this work. Aiming to develop better classifiers, we intend to explore the use of word order and the syntactical and/or semantical information on the representation of documents.

References

1. C. Apté, F. Damerau, and S. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 1994.
2. T. Gonçalves and P. Quaresma. A preliminary approach to the multi-label classification problem of Portuguese juridical documents. In *11th Portuguese Conference on Artificial Intelligence*, Lecture Notes on Artificial Intelligence 2902, pages 435–444, Évora, Portugal, December 2003. Springer-Verlag.
3. Teresa Gonçalves and Paulo Quaresma. The impact of NLP techniques in the multi-label classification problem. In *Intelligent Information Systems 2004*, Advances in Soft Computing, Zakopane, Poland, May 2004. Springer-Verlag. (to appear).
4. N. Japkowicz. The class imbalance problem: Significance and strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000)*, volume 1, pages 111–117, 2000.
5. T. Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers, 2002.
6. K.Nigam, A.McCallum, S.Thrun, and T.Mitchell. Text classification from labelled and unlabelled documents using EM. *Machine Learning*, 39(2):103–134, 2000.
7. D. Mladenić and M. Grobelnik. Feature selection for unbalanced class distribution and Naïve Bayes. In *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 258–267, 1999.
8. P. Quaresma and I. Rodrigues. PGR: Portuguese attorney general's office decisions on the web. In Bartenstein, Geske, Hannebauer, and Yoshie, editors, *Web-Knowledge Management and Decision Support*, Lecture Notes in Artificial Intelligence LNCS/LNAI 2543, pages 51–61. Springer-Verlag, 2003.
9. G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
10. B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.
11. V. Vapnik. *The nature of statistical learning theory*. Springer, 1995.
12. I. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, 1999.

Architecture of a Medical Information Extraction System

Dalila Bekhouche^{1,2}, Yann Pollet³, Bruno Grilheres^{1,2}, and Xavier Denis^{1,4}

¹ EADS DCS, d'Affaires des Portes,
27106 Val de Reuil CEDEX, France

{dalila.Bekhouche, bruno.grilheres}@sysde.eds.net

² PSI Insa Rouen, Place E. Blondel,
76130 Mont St Aignan. France

³ Conservatoire national arts & métiers CNAM,
pollet@cnam.fr

⁴ LIH – Université du Havre
25 rue P. Lebon. 76600 Le Havre
xavier.denis@operamail.com

Abstract. We present in this article an innovative architecture of information extraction system applied to the medical domain. The content of documents (free texts) can be described or reduced to some relevant information. Our aim is to set a process in order to exploit efficiently the content of the documents. We will explain that the medical information extraction task can be analysed into three steps: Extraction “identify and extract a set of events and entities like date, names, medical terms”, Generation “create from these events and entities the relevant information”, Knowledge acquisition “validate and correct the extraction and generation results”. These analysis require to make various approaches in linguistic, statistic and artificial intelligent cooperate and use together specialised terminology as medical nomenclatures ICD-10¹ and CCMA² and linguistic resources.

1 Introduction

In this paper, we focus on the extraction of information from medical documents. These documents are often unstructured or semi structured like medical records (e.g. radiology) and stored in textual form. It is difficult to access and exploit automatically this amount of information because of the variety of content and the specific terminology. The physicians use uncertain expressions and sense modifying, which makes the processing of this information very complex and causes difficulties in understanding for most NLP tools. Our aim is to identify events and information described in medical records which concern the patient (signs, diagnosis, acts, and results). It does not consist in extracting the medical concepts (knowledge) involved in the text

¹ International classification of the diseases

² Common Classification of the medical Acts

(e.g. in a scientific paper) but to precise the instantiated concept resulting from the medical event e.g. “brain tumour”. In this research, we explore several ways of extracting entities and events from the medical records. We represent these documents as vectors of features (i.e. **D_e**: date of the medical event (03/04/04), **D_t**: document type as colonoscopy record, **S**: signs or symptoms as fever, **A**: medical acts, **D**: diagnosis, **R**: results), each feature is associated to a part of the text. The documents are characterised by their compounds parts which can be expressed as symbolic forms e.g. medical terms, dates. To reach this goal, we don't apply deep syntactic and semantic processing like in NLP based approaches, as it would be too complex for our problem and as we need concrete results in operational situations. In the contrary, we develop an original approach combining the reuse of existing resources with simple local syntactic rules.

2 Information Extraction

The information extraction (IE) consists in identifying and extracting relevant information from free natural language text and storing them in relational databases. This task is complex because it requires analysing textual data by searching the meaning of words within an enunciation context. The problem can be reduced to produce factual data by filling predefined template. It consists in identifying references for a predefined set of concepts (companies, proper name) by using rules based on the specific domain vocabulary.

Many systems have been developed to extract information automatically from free texts and are commonly based on pattern matching and are usually designed for specific domains [18]. Most of these systems need special training resources such as AutoSlog [13] that requires a training corpus which is manually tagged with domain specific annotations in order to generate extraction patterns. AutoSlog-TS [8] generate extraction patterns using a pre-classified training corpus but the texts do not need to be annotated. PALKA [12] requires domain specific frames with keyword lists. CRYSTAL [10] requires an annotated training corpus. RAPIER [5] requires filled templates. So as to evaluate those systems, the MUC³ (a DARPA⁴ U.S sponsored conference) aims at comparing automatic extracted information to expert work. The accuracy of the systems are measured in terms of precision, recall and f-measure.

3 Methodology

The structure of medical records depends on the type of examination e.g. a colonoscopy record describes symptoms and acts relating to the intestine and a chemotherapy record contains chemotherapy and scans acts. In general, a medical record describes a

³ Message Understanding Conference

⁴ Defence Advanced Research Projects Agency

set of relevant information related to a patient's examination (for example: practitioners names, patient, location, dates, quantities, modifiers, age, medical vocabulary (fever), tenses and co-references). We propose to use several approaches to process the various phases of extraction. We process texts in sequential steps (see Fig. 1). The first step consists of automatically extracting concepts and relations from texts "annotation" by using lexical resources and rules; the second step automatically generates relevant information "features"; this step requires rules and semantic knowledge. The last step consists in enriching the dictionary and rules under expert controls.

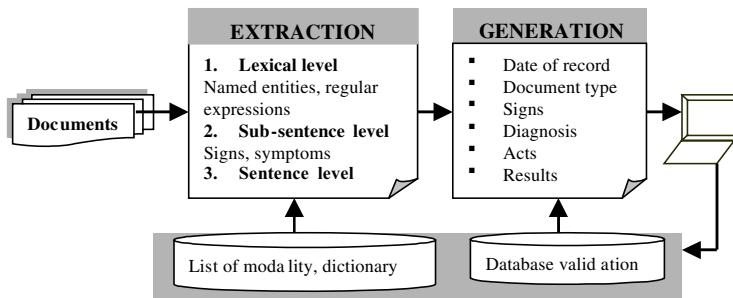


Fig. 1. Architecture for the extraction of medical information

4 Architecture of the EI Medical System

The corpus used in these experiments was created from 8 732 radiological records. It contains 156512 tokens, 4665 verbs, 36300 nouns, 4100 adverbs, and 11813 adjectives. In this section, we describe the architecture of a medical EI system and we process the texts in three hierarchical processing steps.

4.1 Extraction of Events and Relations

In this step, we automatically annotate the text by using dictionaries and rules. We extract the different concepts "date, names" and relations "addressed" into three hierarchical levels.

The first level called the **lexical level** (example1) consists in identifying the general lexical terms. All general information such as named entities are extracted. We apply a name recognition module (lists and regular expressions) to instantiated concepts like location, companies, organizations, modifiers, modalities, first name, and dates.

Example1: *Mr. Smith was addressed for a checkup by McGann*
family name: Regex (word)^{*} or dictionary \Rightarrow *Smith* is a family name

The information obtained in this level will be used in the next levels to identify semantic annotations.

The second level (**sub-sentence level**) is used to detect and extract medical events and semantic annotation. We identify the concepts into a sub-sentence by using semantic rules and thesaurus (example2).

Example2: *Mr. Smith was addressed for a checkup by McGann*

Patient name: Name +Predicate (addressed) + for +Regex(word)^{*} \Rightarrow Smith is the patient and McGann is the practitioner name.

To identify signs, diseases and medical acts, we use the nomenclatures ICD-10; CCMA is used for identifying acts. Our goal is to identify the various occurrences of ICD-10 entries which appear within the text, for that we use a two steps approach: The first step consists in recognizing the occurrences of some predefined strongly discriminate terms, These terms have been previously extracted from the ICD-10, and each of them may index several entries of the ICD-10.

The second step consists in identifying the right ICD-10 concept occurrence. This is done by evaluating the similarity (using the cosine measure in the VSM) between the neighbouring terms and each candidate entry of the ICD-10 in relationship with indexing term. A pre-processing step consists in reducing the dimensionality of texts and thesauri by orthographical correction, standardisation of words, and by removing irrelevant words (articles, determinants) and highest-frequencies term removal. The ICD-10 size has been reduced by 92% (from 17918 diseases and signs down to 1433).

The second process consists in carrying out both syntactic and semantic analysis on these results to improve simple syntactic analysis by introducing tacit knowledge related to a domain. The syntactic analysis will be carried out using a parser, which uses the formalism of lexical tree adjoining grammar (LTAG). A complementary local analysis is performed to extract the sense modifiers and to make a semantic homogenisation. This enables us to weight the first extracted nuance. It may be either a weak or strong hypothesis, or even a negation. Distinctions must be made between semantic modifiers of modality (as expressed above) and types of concept. We detect either a modality (might, may, not) or a nominal group in order to detect if the word “fever” refers to a symptom (this patient has fever) or to a disease (“this patient suffers from yellow fever”).

In this last step called the **sentence level** (example3), we plan to combine the information obtained in levels 1and 2 to induce semantic annotation.

Example3: Mr. Smith was addressed for a checkup by McGann

Examination: Regex (word | annotation level1| annotation level2)

Addressed (patient name, practionners name, examination) \Rightarrow Checkup is an examination.

4.2 Generation and Acquisition

The aim of this paper is not to focus on those two steps, nevertheless we will give you an overview of the Generation and Acquisition process in the following paragraph.

The generation process consists in constructing a prediction sequence described by influent “annotations” and candidate “features”. The idea is to combine previously found annotation to fill in the list of features. Each feature contains several annotations and summarise the relevant information contained in the document. We predefine the rules for each feature as document type, signs, diagnosis, acts, results.

The acquisition process (i.e. the user feedback.) aims at validating and correcting the rules and concepts used in the system. Moreover, validation will allow future corrections on the processing step and to find relations between input information and results of the extraction.

4.3 Validation and Limits of the Approach

We carried out two tests that we describe in the bellow :

The first one corresponds to the evaluation of the system without adding specific knowledge to the system (using the system as it was designed with the reduced CIM10) and the second one is the reduced CIM10 + extra knowledge (= synonyms of terms in the CIM10).

For this evaluation (see Fig.2), we used the typical precision and recall measures adapted to our needs and we defined too new measures that we called the Proposition Ability and the System Interest.

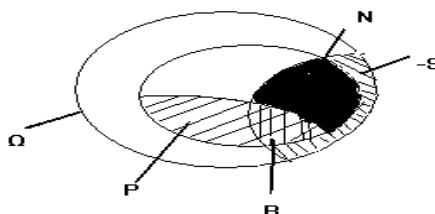


Fig. 2. Methodology of evaluation

Let Ω be the set of all possible annotation of the ICD-10. Let V the set of valid annotation for a document. Let P be the set of valid annotation found by one practitioners. We assume that $\neg P$, set of invalid annotation found by the practitioners, is empty ($\neg P = \emptyset$). Moreover, we assume that he/she has only a focused knowledge on a specific part of the ICD-10, and that he/she is not able to retrieve all correct annotations for a document. As a consequence, $P \subset V$ but not necessarily $P = V$.

Let S be the set of valid annotation found by the system and $\neg S$ be the set of invalid annotation of the system. We define B as the set of valid annotation found both by the system and the practitioners. We also define N the set of new annotations found only by the system $N = S - B$. The precision Pr can be deducted from the previous definition as follow: $Pr = \frac{B}{P}$. The recall Re is $Re = \frac{B}{B + \neg S}$. We define a new measure the **Proposal Ability Pa** that evaluates the capacity of the system to propose correctly

new annotations. $Pa = \frac{N}{N+S}$. We define another measure the **System Interest Si**, that evaluates the capacity of the system to propose valid annotation not found by a practitioners. We also calculate the confidence interval at 95 % by using the formula proposed by Bennani and Bossaert (1996) that corresponds to a gaussian repartition of the error (classic assumption).

Table 1. Results of evaluation

	<i>N. doc</i>	<i>P_r</i>	<i>R_e</i>	<i>P_a</i>	<i>S_i</i>
<i>Before adding knowledge</i>	313	0,500	0,710	0,792	0,609
<i>Confidence interval</i>		0.44;0.55	0.66;076	0.74;0.83	0.55;0.66
<i>After adding knowledge</i>	370	0,877	0,798	0,840	0,905
<i>Confidence interval</i>		0.84;0.91	0.75;0.84	0.8;0.87	0.87;0.93

The evaluation (see table1) presents the result we obtained with the two evaluations. We obtain 50% correct annotations for the first test. After adding knowledge, the precision increases up to 87.7%. The recall is approximately the same for the two evaluations around 75%. Recall score represents problems due to ambiguous words and could depend on the context e.g. “STENOSE” is a disease or a part of the body. The system is sometimes able to find more information than the practitioner as his/she is not aware of all the ICD-10, in the contrary, the system is not able to induce domain knowledge. The Proposal Ability as well as the System Interest increases slightly while adding knowledge.

Conclusion and Future Work

We have described the architecture for an information extraction system applied to the medical domain, which is implemented and will be evaluated. The information extraction from “French medical document” is a complex task because lexical resources, such as lists of words with inflectional and derivational information, medical dictionary are not available. In this paper, we focused in annotating text; we have used many techniques and constructed resources to identify specialized vocabulary. Nevertheless, our approach suffers from several limits:

- This approach has been applied to French medical texts only and tested on a specific domain composed of colonoscopy records.

- The system is not always able to determine the ICD-10 entry but proposes an ordered list of candidates entries. The ability of the system to propose the correct CIM10 annotation entry will be evaluated in near future.
- This approach analyses simple sentences as medical records but may have difficulties to analyse complex sentences needing a deep syntactic analysis.
- In next study, we will focus on the generation and acquisition steps and we will take into account synonyms.

References

1. Appelt D., Israel D.: Introduction to Information Extraction Technology. In a tutorial for IJCAI-1999.
2. Aseltine, J.: WAVE: An Incremental Algorithm for Information Extraction. In Proceedings of the AAAI 1999 Workshop on Machine Learning for Information Extraction
3. Pazienza M.T.: Information extraction, toward scalable systems. In Springer Verlag (1999).
4. Riloff E., Schmelzenbach M.: An Empirical Approach to Conceptual Case Frame Acquisition. In Proceedings of the 6th Workshop on Very Large Corpora. (1998)
5. Califf M.E., Mooney R.J.: Relational Learning of Pattern-Match Rules for Information Extraction. In Proceedings of the ACL Workshop on Natural Language Learning, pages 9-15. (1997).
6. Grishman R.: Information Extraction: Techniques and Challenges. Springer Verlag, pp.10-27. (1997)
7. Fisher D., Soderland S., McCarthy J., Feng F. and Lehnert W.: Description of the Umass Systems as Used for MUC-6., In Proceedings of the 6th MUC, (1996).
8. Riloff E.: Automatically generating extraction patterns from untagged text. In AAAI'96, Portland. Pp.1044-1049. (1996)
9. Soderland S., Aronow D., Fisher D., Aseltine J., Lehnert W.: Machine Learning of Text Analysis Rules for Clinical Records, CIIR Technical Report.(1995)
10. Soderland S., Aronow D., Fisher D., Aseltine J., Lehnert W.: CRYSTAL: Inducing a conceptual dictionary. In Proceedings of the 14th International Joint Conference on AI, pages 1314-1319. (1995)
11. Rassinoux A-M.: Extraction et représentation de la connaissance tirée de textes médicaux. Thèse. Département informatique, université de Genève. 322p.(1994)
12. Kim J., Moldovan D.: Acquisition of Semantic Patterns for Information Extraction from Corpora. In Proceedings of the 9th IEEE Conference on AI for Applications, pages 171-176.(1993)
13. Riloff, E.: Automatically Constructing a Dictionary for Information Extraction Tasks. In Proceedings of the 11th National Conference on Artificial Intelligence (AAAI'93), 811-816. Washington DC.(1993)
14. Fisher D., Riloff E.: Applying Statistical Methods to Small Corpora: Benefiting from a Limited Domain," AAAI Symposium on Probabilistic Approaches to Natural Language, 47-53. (1992)
15. Jackson P. Moulinier I. Natural Language Processing for Online Applications Text Retrieval, Extraction and Categorization. Natural Language Processing, pp.75-111. (1992)

16. Riloff E., Schafer C., Yarowsky D.: Inducing Information Extraction Systems for New Languages via Cross-Language Projection. In Proceedings of 19th International Conference on Computational Linguistics (COLING 2002).
17. Pillet V.: Méthodologie d'extraction automatique d'information à partir de la littérature scientifique en vue d'alimenter un nouveau système d'information: Application à la génétique moléculaire pour l'extraction d'information sur les interactions. Thèse. Spécialité sciences de l'information. Aix-Marseille III.(2000)
18. Yangarber R., Grishman R., Tapanainen P., Huttunen S.: Unsupervised Discovery of Scenario-Level Patterns for Information Extraction. In Applied Natural Language Processing Conference (ANLP): 282-289.(2000).

Improving Information Retrieval in MEDLINE by Modulating MeSH Term Weights^{*}

Kwangcheol Shin and Sang-Yong Han⁺

Computer Science and Engineering Department,
Chung-Ang University, 156-756, Seoul, Korea
kcshin@archi.cse.cau.ac.kr, hansy@cau.ac.kr

Abstract. MEDLINE is a widely used very large database of natural language medical data, mainly abstracts of research papers in medical domain. The documents in it are manually supplied with keywords from a controlled vocabulary, called MeSH terms. We show that (1) a vector space model-based retrieval system applied to the full text of the documents gives much better results than the Boolean model-based system supplied with MEDLINE, and (2) assigning greater weights to the MeSH terms than to the terms in the text of the documents provides even better results than the standard vector space model. The resulting system outperforms the retrieval system supplied with MEDLINE as much as 2.4 times.

1 Introduction

MEDLINE is a premier bibliography database of National Library of Medicine (NLM; www.nlm.gov). It covers the fields of medicine, nursing, dentistry, veterinary medicine, the health care system, the preclinical sciences, and some other areas of the life sciences. MEDLINE contains bibliographic citations and author abstracts from over 4,600 journals published in the United States and in 70 other countries. It has approximately 12 million records dating back to 1966 [11].

Medical Subject Headings (MeSH) is the authority list of controlled vocabulary terms used for subject analysis of biomedical literature at NLM [8]. It provides an extensive list of medical terminology having a well-formed hierarchical structure. It includes major categories such as anatomy/body systems, organisms, diseases, chemicals and drugs, and medical equipment.

Expert annotators of the National Library of Medicine databases, based on indexed content of documents, assign subject headings to each MEDLINE document for the users to be able to effectively retrieve the information that explains the same concept with different terminology. Manual annotation with MeSH terms is a distinctive feature of MEDLINE [11].

^{*} Work supported by the ITRI of the Chung-Ang University.

⁺ Corresponding author

MeSH keywords assigned to each individual document are subdivided into *MeSH Major headings* and *MeSH Minor headings*. MeSH Major headings are used to describe the primary content of the document, while MeSH Minor headings are used to describe its secondary content. On average, 5 to 15 subject headings are assigned per document, 3 to 4 of them being major headings [8].

MEDLINE is supplied with its own search engine. To use it, users give their keywords as a query to the system. The system automatically converts the query into Boolean form and retrieves the data from the MeSH field and the author information fields. No relevance ranking is provided; the retrieved documents are returned in no particular order.

We show that applying a vector space model-based search engine [3] to full-text MEDLINE data gives much better results. More importantly, we show that assigning greater weights to the MeSH terms than to the words from the full text of the document further improves the quality of the results. In this way, we obtain better ranking of the search result than with the traditional vector space model which used in the SMART system [13] and much better results than with the Boolean model used in the search engine provided with MEDLINE.

This paper is organized as follows. Section 2 gives a short introduction to the vector space model. Section 3 describes the proposed technique to modulate the MeSH terms weights. Section 4 presents the experimental results. Finally, Section 5 provides some conclusions.

2 Vector Space Model

In the vector space model [13] the documents are represented as vectors with the coordinates usually proportional to the number of occurrences (*term frequency*) of individual content words in the text. Namely, the following procedure is used for converting documents into vectors [3, 4]:

- Extract all the terms from all n documents in the collection;
- Calculate the frequency f_{ij} of the terms i for each document j ;
- Exclude the terms without semantic meaning and the terms with very high or very low frequency (such excluded terms are called *stopwords*);
- Calculate the number n_i of the documents where each term i occurs;
- Allocate indices from 1 to d to each remaining term, where d is the number of such terms. Allocate indices from 1 to n to each document. The vector space model for the entire document collection is determined by the $d \times n$ -dimensional matrix $w = \|w_{ij}\|$, where w_{ij} usually refers to the *tf-idf* (*term frequency-inverse document frequency*) value of the i -th term in j -th document calculated as

$$tf\text{-}idf = \frac{f_{ij}}{\max f_{ij}} \log \frac{n_i}{n} \quad (1)$$

After converting each document to a vector, we can measure the similarity between two documents (vectors) using the cosine measure widely used in information

retrieval: the cosine of the angle between the two vectors. This measure is easy to understand and its calculation for sparse vectors is very simple [4]. Specifically, the cosine measure between the user query and a document is used to quantitatively estimate the relevance of the given document for the given query.

The cosine similarity between two vectors x_i and x_j is their inner product:

$$s(x_i, x_j) = \frac{x_i^T x_j}{\|x_i\| \|x_j\|} = \cos(\theta(x_i, x_j)),$$

where θ is the angle between the two vectors. To simplify calculations in practice, the vectors are usually normalized so that their norm $\|x\|$ be 1. If the two documents have no word in common, the similarity is 0. On the contrarily, the similarity between two copies of same document is 1. If the two documents have some words in common, their similarity is between 0 and 1.

The vector space model has the advantage over the Boolean model in that it provides relevance ranking of the documents: unlike the Boolean model which can only distinguish relevant documents from irrelevant ones, the vector space model can indicate that some documents are very relevant, others less relevant, etc.

3 Modulating MeSH Term Weights

MEDLINE documents contain MeSH keywords as shown in Fig. 1. The MJ field lists the major MeSH terms manually assigned to this document, MN field the minor MeSH terms, and AB the full text of the document—the abstract of a research paper.

MJ	BONE-DISEASES-DEVELOPMENTAL: co.	CYSTIC-FIBROSIS:			
	co.	DWARFISM: co.			
MN	CASE-REPORT.	CHILD.	FEMALE.	HUMAN.	SYNDROME.
AB	Taussig et al reported a case of a 6-year-old boy with the Russell variant of the Silver-Russell syndrome concomitant with cystic fibrosis. We would like to describe another patient who...				

Fig. 1. A sample of MEDLINE data

Our idea is to increase the weight of MeSH terms in each document vector, since these terms are known to be more important than other words in the text of the document.

Apart from our experimental results supporting our suggestion, it can be justified in the following way. A MeSH keyword assigned by the reviewer “stands for” several words in the document body that “vote” for this more general keyword. For example, for the text “... *the patient is allergic to ... the patient shows reaction to ... causes itch in patients ...*” the annotator would add a MeSH term ALLERGY. Though this term appears only once in the document description, it “stands for” three matching terms in the text body, namely, *allergic*, *reaction*, and *itch*. Our hypothesis is that increasing

its weight would more accurately describe the real frequency of the corresponding concept in the document and thus lead to better retrieval accuracy.

We use the following procedure. First, we assign the weights as described in the previous section. Then we use the formula (2) to increase the weight of MeSH terms:

$$w_{ij} \leftarrow \begin{cases} w_{ij} + \left(\rho + \frac{\rho}{4 + \ln(\rho)} \right) & \text{if the term } j \text{ is MeSH Major in document } i \\ w_{ij} + \left(\rho - \frac{\rho}{4 + \ln(\rho)} \right) & \text{if the term } j \text{ is MeSH Minor in document } i \end{cases} \quad (2)$$

where ρ is a parameter regulating the sensitivity of the formula to the MeSH terms. The specific expressions in (2) were found empirically. The formula adds slightly different values to the terms according to their significance. E.g., with $\rho = 0.3$ additional term weights are $0.3 + 0.017 = 0.317$ and $0.3 - 0.017 = 0.283$, respectively. This is because MeSH major terms are more important than MeSH minor terms.

4 Experimental Results

We experimented with the well-known Cystic Fibrosis (CF) reference collection, which is a subset of MEDLINE. It has 1,239 medical data records supplied with 100 queries with relevant documents provided for each query. A sample query is shown in Figure 3. The 3-digit numbers are the numbers of the documents known to be relevant for this query. The four-digit groups reflect the relevance scores ranging from 0 to 2 assigned to the given document by four experts who manually evaluated the relevance of each document with respect to each query.

QU	What are the effects of calcium on the physical properties of mucus from CF patients?
RD	139 1222 151 2211 166 0001 311 0001 370 1010 392 0001 439 0001 440 0011 441 2122 454 0100 461 1121 502 0002 503 1000 505 0001

Fig. 2. An example of a CF query with answers.

We used the MC program [2] to convert documents from the CF collection into vectors. Stopwords and the terms with frequency lower than 0.2% and higher than 15% were excluded. With this, the CF collection had 3,925 terms remaining. Then the *tf-idf* value was calculated for each document according to (1) and then the obtained 1,239 vectors were normalized.

To verify the proposed method, the MeSH terms were extracted from the documents and their term weights in each document vector were modulated by (2). Then the vectors were re-normalized. The test result with different additional term weights for MeSH keywords are shown in Figure 3. Two plots are presented:

- the average R-Precision (relative number of relevant documents among the R highest-ranked ones) on 100 queries, and
- the total relevance scores of the R highest-ranked documents,

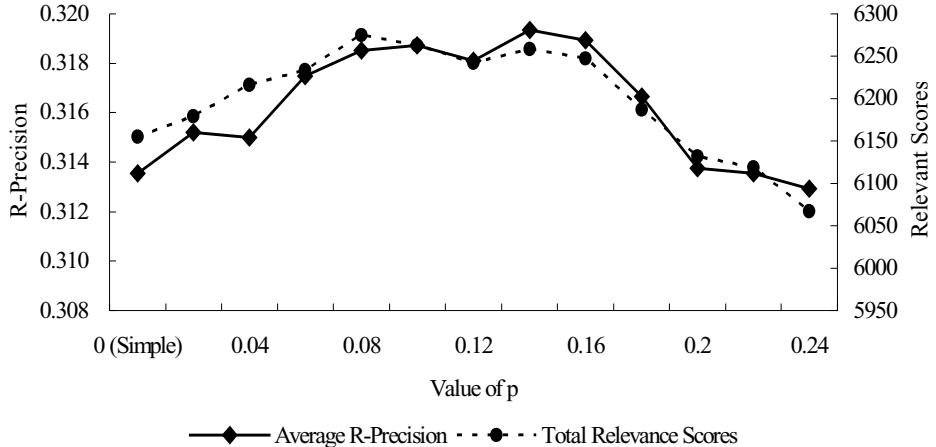


Fig. 3. Experimental results with different parameter ρ .

where R is the total number of relevant documents in the collection for a given query. We considered a document relevant if at least one on the four experts judged it relevant, see Figure 2. By the score of a document we refer to the scores assigned to it by the four experts, which ranges from 1 (e.g., 1000 in Figure 2) to 8 (2222).

Note that the additional value of 0 corresponds to the baseline: vector space model without modulating the MeSH terms' weights, as used in the SMART system [13].

The additional weight of 0.14 assigned to MeSH terms gives 1.85% of improvement in R -Precision as compared with the baseline ($\rho = 0$). The total relevance scores reach their maximum of 1.95% of improvement over the baseline when the additional weight is 0.08. Note that the results are not very sensitive to a specific value of ρ , around the value of $\rho = 0.1$.

To further clarify the effect of modulating the MeSH terms' weights, we tested our system taking into account only abstracts or both abstracts and MeSH terms (fields AB, MJ, and MN in Figure 1). Also, we tried stemming [12], with which the traditional vector space model shows better results than on non-stemmed data. Namely, on the abstracts with MeSH terms, stemming gave about 10.26% of improvement for the traditional vector space model and 12.57% for the suggested method (modulating the MeSH terms' weights). Table 1 shows that the best performance is achieved on modulated MeSH terms' weights with stemming, while the worst performance on the traditional vector space model with only abstracts (no MeSH terms at all).

To compare our results with the Boolean model currently used in the search engine provided with MEDLINE, we extracted the MeSH terms from each query and searched the documents with same MeSH terms using the OR operation (if the AND operation is used, no documents are usually retrieved).

Table 1. Experimental results.

	Vector (abstract only)	Vector (abstract and MeSH)	Suggested Method ($\rho=0.14$)	Vector with Stemming	Suggested Method with stemming ($\rho=0.10$)
R	4819	4819	4819	4819	4819
$\#R$	1343	1511	1537	1666	1701
T	5358	6155	6279	6951	7063
R -precision $= \#R / R$	0.279	0.314	0.319	0.346	0.353

R is the number of correct documents in the collection;

$\#R$ is the number of the relevant documents among the R highest-ranked ones;

T is the total scores of the R highest-ranked documents.

Test result is shown in Table 2. For the 100 queries provided with the CF collection, a total of 17155 documents are retrieved, of which only 1783 are relevant (no stemming was applied). In case of the suggested method, however, of the highest-ranked $R = 4819$ documents, 1583 ones are relevant. So the suggested method gives as much as 2.4 times improved precision/recall ratio.

Table 2. Experimental results. Notation is as in Table 1.

	Boolean	Boolean with stemming	Suggested method with stemming ($\rho = 0.10$)
R	17155	23149	4819
$\#R$	1783	2203	1701
T	6672	8114	7063
Precision or R -precision	0.104	0.095	0.353

5 Conclusions and Future Work

We have shown that taking into account both full texts of the documents and the MeSH terms in frame of the vector space model gives much better retrieval results on the MEDLINE data than the Boolean model-based system currently provided with MEDLINE, which takes into account only the MeSH terms.

What is more important, we have shown that increasing the weight for MeSH terms as compared with the standard vector space model improves retrieval accuracy. We applied different increasing coefficients to major and minor MeSH terms. With

the best combination of the parameters (vector space model, stemming, ≈ 0.1) we obtained as much as 2.4 times better results than the system currently provided with MEDLINE. Our experimental results show that the method is not very sensitive to a specific value of the parameter .

In the future we plan to investigate the effects of automatic learning individual weights for each MeSH term instead of a common parameter . In particular, we plan to automatically determine the weight of MeSH term for the given document (or even automatically assign new MeSH terms to the documents) in the way suggested in [6]. With this, we expect to be able to use other hierarchical vocabularies, such as WordNet, to index the documents in MEDLINE. The selection of terms for indexing will be done according to the technique described in [5].

Finally, we plan to try semantically rich representations of the text structure, such as conceptual graphs [10]. As an intermediate step to this direction, we will apply shallow pattern-based analysis to the titles of the documents stored in MEDLINE [9].

References

1. Dhillon I. S. and Modha, D. S. *Concept Decomposition for Large Sparse Text Data using Clustering*. Technical Report RJ 10147(9502), IBM Almaden Research Center, 1999.
2. Dhillon I. S., Fan J., and Guan Y. Efficient Clustering of Very Large Document Collections. *Data Mining for Scientific and Engineering Applications*, Kluwer, 2001.
3. Baeza-Yates, R., and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.
4. Frakes W. B. and R. Baeza-Yates. *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.
5. Gelbukh, A. Lazy Query Enrichment: A Simple Method of Indexing Large Specialized Document Bases. In Proc. DEXA-2000, Database and Expert Systems Applications. *Lecture Notes in Computer Science*, N 1873, Springer, pp. 526–535.
6. Gelbukh, A., G. Sidorov, and A. Guzmán-Arenas. A Method of Describing Document Contents through Topic Selection. In Proc. SPIRE'99, *String Processing and Information Retrieval*, IEEE Computer Society Press, 1999, pp. 73–80.
7. Ide E. New experiments in relevance feedback. In G. Salton, editor, *The SMART Retrieval System*, 337–354, Prentice Hall, 1971.
8. Lowe H.J., Barnett O. Understanding and using the medical subject headings (MeSH) vocabulary to perform literature searches. *J. American Medical Association*, 1995; 273:184.
9. Montes-y-Gomez, M., A. López López, A. Gelbukh. Document Title Patterns in Information Retrieval. In Proc. TSD-99, Text, Speech and Dialogue. *Lecture Notes in Artificial Intelligence*, N 1692, Springer, 1999, pp. 364–367.
10. Montes-y-Gomez, M., A. López López, A. Gelbukh. Information Retrieval with Conceptual Graph Matching. In Proc. DEXA-2000, Database and Expert Systems Applications. *Lecture Notes in Computer Science*, N 1873, Springer, 2000, pp. 312–321.
11. MEDLINE Fact Sheet. www.nlm.nih.gov/pubs/factsheets/medline.html.
12. Porter, M. An algorithm for suffix stripping. *Program*, 14, 1980, pp. 130–137.
13. Salton G. and. McGill M. J., *Introduction to Modern Retrieval*. McGraw-Hill, 1983.

Identification of Composite Named Entities in a Spanish Textual Database*

Sofía N. Galicia-Haro¹, Alexander Gelbukh^{2,3}, and Igor A. Bolshakov²

¹ Faculty of Sciences
UNAM Ciudad Universitaria México, D. F.
sngh@fciencias.unam.mx

² Center for Computing Research
National Polytechnic Institute, Mexico City, Mexico
 {gelbukh,igor}@cic.ipn.mx; www.Gelbukh.com

³ Department of Computer Science and Engineering, Chung-Ang University,
221 Huksuk-Dong, DongJak-Ku, Seoul, 156-756, Korea

Abstract. Named entities (NE) mentioned in textual databases constitute an important part of their semantics. Lists of those NE are an important knowledge source for diverse tasks. We present a method for NE identification focused on composite proper names (names with coordinated constituents and names with several prepositional phrases.) We describe a method based on heterogeneous knowledge and simple resources, and the preliminary obtained results.

1 Introduction

Textual databases have been moved from desks to computers and also to the Web for many reasons: to save tons of paper, to allow people to have remote access, to provide much better access to texts in an electronic format, etc. Searching through this huge material for information of interest is a high time consuming task.

Named entities (NE) mentioned in textual databases constitute an important part of their semantics and of their lexical content. From a collection of political electronic texts we found that almost 50% of the total sentences contains at least one NE. This percentage shows the relevance of NE identification and its property to be used to index and retrieve documents. In [5] authors employed proper names for an automatic newspaper article classification. The quantity of proper names and their informative quality in such type of texts make them relevant to improve the clustering thanks to a measure of similarity that highlights them with regard to the other words in a text.

The research fulfilled in the Message Understanding Conference (MUC) [8] structure entity name task and it distinguishes three types: ENAMEX, TIMEX and NUTEMEX [4]. In this work we are concerned with ENAMEX that considers entities such as organizations, persons, and localities. Name entity recognition (NER) works in

* Work done under partial support of Mexican Government (CONACyT, SNI, COFAA-IPN), Korean Government (KIPA Professorship for Visiting Faculty Positions in Korea), and ITRI of CAU. The second author is currently on Sabbatical leave at Chung-Ang University.

Language-Independent NER, the shared task of CoNLL-2002 [10] covered Spanish for name entity classification. However, composite names were limited.

In this paper, we are not concerned with classification but with identification of NE focusing our work on composite NE: names with coordinated constituents and names with several prepositional phrases. Since NE recognition is a difficult task our method is heterogeneous; it is based on local context, linguistic restrictions, heuristics and lists for disambiguation (two very small lists of proper names, one of similes, and a list of non ambiguous entities taken from the textual database itself). In this article, we present the text analysis carried out to determine the occurrence of NE, then we detailed our method and finally we present the obtained results.

2 Named Entities in Textual Databases

Textual databases could contain a great quantity of NE; most of them are unknown names. Since NE belong to open class of words, entities, as commercial companies are being created daily, unknown names are becoming important when the entities they referred to became topical or fashioned.

Our textual database is a collection of political Mexican texts that were compiled from the Web. They correspond to two different Mexican newspapers (1998 to 2001). We called them: collec#1 (442,719 total sentences, 243,165 with NE) and collec#2 (208,298 total sentences, 100,602 with NE). Although NE represent at most 10% of total words of our textual database they appear at least in 50% of the sentences.

Composite NE are common in Spanish texts, for example, the political texts of January 3rd 2000 contain among other NE the following composite names¹:

6	Comandancia General del Ejército Zapatista de Liberación Nacional
6	Comité Clandestino Revolucionario Indígena
2	Comité Clandestino Revolucionario Indígena de la Comandancia General del Ejército Zapatista de Liberación Nacional
8	Ejército Zapatista de Liberación Nacional

Since we could observe that *Ejército Zapatista de Liberación Nacional*, *Comandancia General*, and *Comité Clandestino Revolucionario Indígena* are embedded in composite NE the elementary names are more than previous ones:

8	Comandancia General
7	Comité Clandestino Revolucionario Indígena
16	Ejército Zapatista de Liberación Nacional

Obtaining the real quantities of elementary NE should improve different tasks as texts classification.

Characteristics for Named Entities Identification

The initial step for NE recognition was identification of linguistic and style characteristics. We analyzed collec#1 and we found that NE are introduced or defined by means of syntactic-semantic characteristics and local context. The main characteristics observed were:

¹ Where “Ejército” means Army, “Comandancia General” means General Command, “Comité Clandestino Revolucionario Indígena” means Revolutionary Secret committee Native, and “Ejército Zapatista de Liberación Nacional” means Army Zapatista of National Liberation

Linguistic. NE could include: a) conjunctions: “y”, “e” (*Centro de Investigaciones y Seguridad Nacional* is a single NE), b) prepositions (*Comisión para la Regularización de la Tenencia de la Tierra*). NE could be separated by a) punctuation marks (*Misantla, Chicantepec, Veracruz, Salina*), b) prepositions (*Salina Cruz a Juchitán*).

Style. It considers: a) information obtained from juxtaposition of NE and acronyms, for ex: *Partido de la Revolución Democrática (PRD)*, b) introducing NE by specific words, for ex: *dirigentes de la Central Nacional de Estudiantes Democráticos (leaders of the ...)*, and c) diverse forms, for ex: *Centro de Investigación y Estudios Superiores de Antropología Social, Centro de Investigación y Estudios Superiores en Antropología Social*, correspond to the same entity, and more variety exists for NE translated from foreign languages.

3 Named Entities Analysis

We built a Perl program that extracts groups of words that we call “compounds”; they really are the contexts when NE could appear. The compounds contain no more than three non-capitalized words between capitalized words. We supposed that they should correspond to functional words (prepositions, articles, conjunctions, etc.) in composite NE. The compounds are left and right limited by a punctuation mark and a word if they exist. For example, for the sentence: *Esa unidad será dirigida por un Consejo Técnico que presidirá Madrazo Cuéllar y en el que participará el recién nombrado subprocurador Ramos Rivera.* (That unit will be directed by a Technical Advice whom Madrazo Cuéllar will preside over and in which the just named assistant attorney general Ramos Rivera will participate.) We obtained the following compounds:

- *por un Consejo Técnico que presidirá Madrazo Cuéllar y*
- *subprocurador Ramos Rivera.*

From 243,165 sentences 472,087 compounds were obtained from collec#1. We analyzed 500 sentences randomly selected and we encountered the main problems that our method should cope with. They are described in the following sections.

Syntactic Ambiguity

- *Coordination.* Singular conjunction cases (“word conjunction word”) cover most of coordinated NE. For ex. *Hacienda y Crédito Público* is a single NE. However, there are cases where the coordinated pair is a sub-structure of the entire name, for ex: *Mesa de [Cultura y Derechos] Indígenas* (Meeting of Culture and Right Natives). Coordination of coordinated NE introduces ambiguity to determine single NE, for example, the group *Comercio y Fomento Industrial y Hacienda y Crédito Público* contains two organization NE where the second one is underlined.
- *Prepositional phrase attachment (PPA)* is a difficult task in syntactic analysis. NE identification presents a similar problem. A specific grammar for NE is not a solution since it should cope with the already known PPA. We consider a diverse criterion than that considered in CoNLL: in case a named entity is embedded in another name entity or in case a named entity is composed of several entities all the components should be determined. For example: *Centro de Investigación y Estudios Avanzados del Politécnico* (Polytechnic's Center of ...) where *Centro de*

Investigación y Estudios Avanzados is a research center of a superior entity (Polytechnic).

Discourse Structures

Discourse structures could be another source for knowledge acquisition. Entities could be extracted from the analysis of particular sequences of texts. We consider:

- *Enumeration* can be easily localized by the presence of similar entities, separated by connectors (commas, subordinating conjunction, etc). For example, *Cocotitlán, Tenango del Aire, Temamatlla, Tlalmanalco, Ecatzingo y Atlautla*
- *Emphasizing* words or phrases by means of quotation marks. For ex: “*Roberto Madrazo es el Cuello*”, “*Gusano Gracias*”, are parodies of well known names.

4 Method

We conclude on our analysis that a method to identify NE in the textual database should be based mainly on the typical structure of Spanish NE themselves, on their syntactic-semantic context, on discourse factors and on knowledge of specific composite NE. Then, our method consists of heterogeneous knowledge contributions.

We avoid complex methods and big resources. For example, in [9] three modules were used for name recognition: List lookup for names and cues, POS tagger and Name parsing, and Name-matching (against all unidentified sequences of proper nouns). Other systems use lists of names of very different sizes, from 110,000 names (MUC-7) to 25,000-9,000 names [6]. [7] experimented with different types of lists.

The lists of names used by NER systems have not generally been derived directly from text but have been gathered from a variety of sources. For example, [2] used several name lists gathered from web sites. We also included lists from Internet and a hand made list of similes [1] (stable coordinated pairs) for example: *comentarios y sugerencias, noche y día, tarde o temprano*, (comments and suggestions, night and day, late or early). This list of similes was introduced to disambiguate some coordinated groups of capitalized words. The lists obtained from Internet were: 1) a list of personal names (697 items), 2) a list of the main Mexican cities (910 items) included in the list of telephone codes.

Linguistic knowledge. It considers preposition use, POS of words linking groups of capitalized words, and punctuation rules. The linguistic knowledge is settled in linguistic restrictions. For example:

1. Lists of groups of capitalized words are similar entities. Then an unknown name of such lists has similar category and the last one should be a different entity coordinated by conjunction. For example: *Santo Domingo, Granada, Guatemala y Haití*.
2. Preposition “*por*” followed by an undetermined article cannot link groups of person names. The compound: *Cuauhtémoc Cárdenas por la Alianza por la Ciudad de México* must be divided in *Cuauhtémoc Cárdenas* and *Alianza por la Ciudad de México*. Therefore, the last compound could correspond to a single name.

Table 1. Results in a testing set of sentences

	NUMBER OF:		
	COORDINATED GROUPS	PREPOSITIONAL PHRASE GROUPS	ALL
Precision	54	69	89
Recall	48	67	87

Heuristics. Some heuristics were considered to separate compounds. For example:

1. Two capitalized words belonging to different lists must be separated. For example: “...en *Chetumal Mario Rendón* dijo ...”, where *Chetumal* is an item of main cities list and *Mario* is an item of personal names list.
2. One personal name should not be coordinated in a single NE. For ex: *José Ortega y Gasset y Manuel García Morente*, where *Manuel* is an item of personal name list.

Statistics. From collec#1 we obtained the statistics of groups of capitalized words, from one single word to three contiguous words, and groups of capitalized words related to acronyms. The top statistics were used to disambiguate compounds joined by

- *Functional words*. For ex. the compound *Estados Unidos sobre México* could be separated in *Estados Unidos* (2-word with high score) and *México*.
- *Embedded names*. For example: *Consejo General del Instituto Federal Electoral* could be separated in: *Consejo General* and *Instituto Federal Electoral*.

Application of the Method

Perl programs were built for the following steps to delimit NE:

First step: All composite capital words with functional words are grouped in one compound. We use a dictionary with part of speech to detect functional words.

Second step: Using the resources (statistics of collec#1 and lists), rules and heuristics above described the program decides on splitting, delimiting or leaving as is each compound. The process is 1) look up the compound in the acronym list, 2) decide first on coordinated groups, then on prepositional phrases, and finally decide on the rest of groups of capitalized words.

5 Results

Since collec#1 was used for training we use 500 sentences randomly selected of collec#2 to test the method. They were manually annotated and compared. The composite NE were split and each individual NE was annotated. The correct entities detected should have the same individual NE. The results are showed in Table 1 where:

Precision: # of correct entities detected / # of entities detected

Recall: # of correct entities detected / # of entities manually labeled (*eml*)

The table indicates the performance for coordinated names (63 *eml*), prepositional groups² (167 *eml*). The last column shows the overall performance (1496 *eml*) including the previous ones. The main causes of errors are: 1) foreign words, 2) personal names missing in the available list, and 3) names of cities.

² Where all prepositional phrases related to acronyms were not considered in this results.

The overall results obtained by [3] in Spanish texts for name entity recognition were 92.45% for precision and 90.88% for recall. But test file only includes one coordinated name and in case a named entity is embedded in another name entity only the top level entity was marked. In our work the last case was marked incorrect. The worst result was that of NE with coordinated words that should require enlargement of current sources. The 40% of coordinated correct entities detection was based on the list of similes that could be manually enlarged.

Conclusions

In this work, we present a method to identify and disambiguate groups of capitalized words. We are interested in minimum use of complex tools. Therefore, our method uses extremely small lists and a dictionary with POS. Since limited resources use cause robust and velocity of execution.

Our work is focused on composite NE (names with coordinated constituents and names with several prepositional phrases) to obtain elementary NE that are useful for different tasks like texts classification. The strategy of our method is the use of heterogeneous knowledge to decide on splitting or joining groups with capitalized words. The results were obtained on 500 sentences that correspond to different topics. The preliminary results show the possibilities of the method and the required information for better results.

References

1. Bolshakov, I. A., A. F. Gelbukh, and S. N. Galicia-Haro: Stable Coordinated Pairs in Text Processing. In Václav Matoušek and Pavel Mautner (Eds.). *Text, Speech and Dialogue. Lecture Notes in Artificial Intelligence*, N 2807, Springer-Verlag (2003) 27–35
2. Borthwick et al. Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition Proceedings of the Sixth Workshop on Very Large Corpora (1998)
3. Carreras, X., L. Márques and L. Padró. Named Entity Extraction using AdaBoost In: Proceedings of CoNLL-2002, Taipei, Taiwan (2002) 167-170
4. Chinchor N.: MUC-7 Named Entity Task Definition (version 3.5). http://www.itl.nist.gov/iaui/894.02/related/projects/muc/proceedings/muc_7_toc.html#appendices (1997)
5. Friburger, N. and D. Maurel.: Textual Similarity Based on Proper Names. Mathematical Formal Information Retrieval (MFIR'2002) 155–167
6. Krupka, G. and Kevin Hausman. Description of the NetOwl(TM) extractor system as used for MUC-7. In Sixth Message Understanding Conference MUC-7 (1998)
7. Mikheev A., Moens M., Grover C.: Named Entity Recognition without Gazetteers. In Proceedings of the EACL (1999)
8. MUC: Proceedings of the Sixth Message Understanding Conference. (MUC-6). Morgan Kaufmann (1995)
9. Stevenson, M. & Gaizauskas R.: Using Corpus-derived Name List for name Entity Recognition In: Proc. of ANLP, Seattle (2000) 290-295
10. Tjong Kim Sang, E. F.: Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings of CoNLL-2002, Taipei, Taiwan (2002) 155-158

ORAKEL: A Natural Language Interface to an F-Logic Knowledge Base

Philipp Cimiano

Institute AIFB, University of Karlsruhe

Abstract. In this paper we present ORAKEL, a natural language interface which translates wh-questions into logical queries and evaluates them with respect to a given knowledge base. For this purpose, ORAKEL makes use of a compositional approach in order to construct the semantics of a wh-question. The system is in principle able to deal with arbitrary logical languages and knowledge representation paradigms, i.e. relational models, frame-based models, etc. However, in this paper we present a concrete implementation based on F-Logic and Ontobroker as underlying inference engine.

1 Introduction

For many applications it is desirable to query knowledge stored in a database or knowledge base through a natural language interface. Actually, this is an important research problem which has received special attention in the mid 80's (see [1] or [6] for two good surveys of the field). Certainly it is possible to query a knowledge base by using some logical query language, but it is not feasible to assume that non-computer-scientists will find such a language intuitive to use. Another option is to make use of boolean queries such as those used in WWW query interfaces as for example Google¹ or Altavista². This sort of queries are certainly much more intuitive than logical ones, but suffer from a very reduced expressiveness. In this sense the challenge is to use an expressive logical query language in the background while at the same time hiding the complexity of such a language to the user by allowing him to formulate queries in natural language. In this paper we present ORAKEL, a natural language interface for a knowledge base which implements a compositional semantics approach in order to translate wh-questions into logical form (compare [3]). In particular, motivated by the growing importance of object-oriented database systems, we present a translation into F(rame)-logic [12]. F-Logic is a fully-fledged first order logic with a model-theoretic semantics. The logic was originally defined to account for the logical properties of object-oriented systems such as frames, inheritance etc. As underlying F-Logic inference engine we make use of Ontobroker [7].

The remainder of this paper is organized as follows: Section 2 presents the architecture of the system and describes its main components. In Section 3 we

¹ <http://www.google.de>

² <http://www.altavista.com>

show some results of the lexicon generation component. Section 4 concludes the paper.

2 System Architecture and Main Components

The main features of ORAKEL are on the one hand that it makes use of a compositional semantic construction approach as in the JANUS question answering system ([10]) thus being able to handle questions involving quantification, conjunction and negation in a classical way. On the other hand, ORAKEL automatically generates the lexicon needed to interpret the wh-questions from the knowledge base itself. In this respect it differs from earlier systems in which either general-purpose lexicons were used ([2]), developed by the interface engineer or database expert with support of tools as in TEAM ([9]) or developed and incrementally refined through interaction with the users as in RENDEZVOUS ([5]). The architecture of ORAKEL is depicted in Figure 1. In brief, the user asks a wh-question which is parsed by the *parsing and semantic construction component* which uses the *general* and *domain* lexicons as resources. The latter is automatically generated out of the F-Logic knowledge base by a *lexicon generation component*. The resulting F-Logic query is sent to the *Ontobroker* inference engine ([7]) which evaluates the query and directs the answer to the user. The two core components, i.e. the parsing and semantic construction as well as the lexicon generation component are described in the following sections.

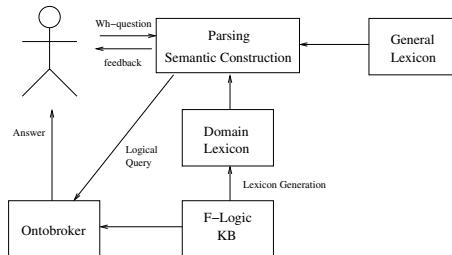


Fig. 1. System Architecture

2.1 Parsing and Semantic Construction

In order to translate wh-questions into F-logic queries we make use of a compositional semantic construction approach presented in [3]. As underlying syntactic theory we build on a variant of *Lexicalized Tree Adjoining Grammar* (LTAG) [11] introduced in [13]. LTAG is especially interesting in this context because of its extended domain of locality and thus the natural way in which subcategorization is treated. However, we extend the formalism to also include ontological information (compare [4]). For further details about the approach to semantic construction the reader is referred to [3]; in this paper we focus in particular on the system's lexicon generation component.

2.2 Lexicon Generation

An important question for any natural language interface and in general any syntax-semantics interface is where the necessary lexical entries come from. ORAKEL is novel in this respect in the sense that it automatically generates the lexicon - i.e. the elementary trees - out of the knowledge-base in question. First of all, it is important to mention that the parser makes use of two different lexicons: the *general lexicon* and the *domain lexicon* (compare Figure 1). The general lexicon includes closed-class words such as determiners i.e. *the*, *a*, *every*, etc., as well as question pronouns, i.e. *who*, *what*, *which*, *where*, etc. and thus is domain independent. The domain lexicon varies from application to application and is generated out of the knowledge base. For this purpose, ORAKEL makes use of subcategorization information automatically acquired from a big corpus, in our case the British National Corpus (BNC). In particular, we parse the corpus with LoPar ([15]), a statistical left-corner parser and extract the following syntactic frame types: intransitive, transitive, intransitive+PP, transitive + PP for verbs and N+PP and N+PP+PP for nouns. For each verb or noun and for each frame, we then take the synset from WordNet ([8]) which best generalizes the selectional preferences at each argument position in line with [14].

At a second step, we then take each method name in the knowledge base and look for a subcategorization frame with the same arity. We then check if the concepts are compatible, i.e. if there is a mapping M from the arguments in the subcategorization frame to the ones of the F-Logic method signature such that the concepts specified in the method are more special than the ones in the subcategorization frame with regard to the WordNet lexical hierarchy. Out of these compatible subcategorization frames we choose the one maximizing the product $\frac{1}{\sum_{(i,j) \in M} \Delta_{WN}(c(i), c(j))} \times p(s_v|v)$, where $(i, j) \in M$ says that position i of the subcategorization frame has been mapped to position j of the method signature and $c(i)$ is the value of the concept at position i in the method and $c(j)$ is the ID of the WordNet synset best generalizing the argument at position j in the subcategorization frame. Further, $\Delta_{WN}(c(i), c(j))$ is then the distance with regard to WordNet between the two concepts and $p(s_v|v)$ is simply the conditional probability of the frame s_v given the predicate (verb) v . In fact, there are different syntactic frames with an appropriate arity for the method *own[person \Rightarrow company]* for example, so we take the one with the mapping maximizing the above product, i.e. the transitive use of *own* in this case. The synset IDs of the subject and object position both refer to the synset to which *entity* and *something* belongs. As *person* and *company* are both hyponyms of this synset, it is a valid syntactic frame according to our method. In this particular case, the overall distance remains the same independently of which frame argument is mapped to which method argument, so that we keep the argument order thus mapping the subject to the first and the object to the second position of the *own*-method. Then, we automatically generate the corresponding elementary trees to be able to ask for the subject as in *Who owns every company?*, the object as in *What/Which company does Bill Gates own?* as well as both, i.e. *Who owns what?*. In addition, we also generate the passive forms of *own* such that

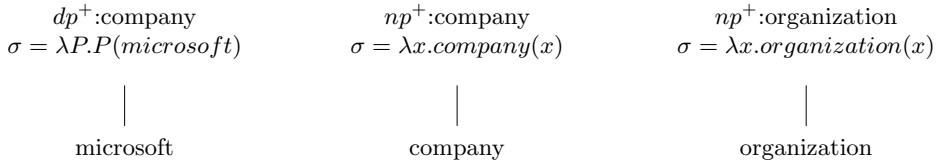


Fig. 2. Elementary trees in the domain lexicon

we can also ask: *What/Which company is owned by Bill Gates?* or *What/Which company is owned by whom?*. For this purpose, we look up the derived forms of *own* (past participle, 3rd person singular present tense) in the lexicon provided with LoPar.

Additionally, for the method names we look up the WordNet-synonyms of the most frequent sense and repeat the same procedure as above for the synonyms in WordNet. For *own*, WordNet contains two synonyms in the most frequent sense, i.e. *have* and *possess*. As expected, the corresponding subcategorization frames are quite similar to the ones of *own*, such that we automatically create similar elementary trees as above for *have* and *posses* mapping to the *own*-method.

Though the mapping as well as generation of elementary trees for method names corresponding to a noun - for example *boss* - works basically along these lines, there is one further complication to take into account. In fact, for such type of methods, there will always be one argument position which is not realized in a noun+PP or noun+PP+PP frame as it is the argument which is typically specified in a copula construction such as *The boss of Microsoft is Bill Gates*. Thus, if the method name corresponds to a noun as in *company[boss ⇒ person]*, we search for compatible frames without considering the last method argument and choose the mapping maximizing the above product. Take for example the *boss* method. As synonyms in WordNet we find: *foreman*, *chief*, *gaffer*, *honcho*. If we then apply our approach we map the subcategorization frames *boss(of:100001740)* and *chief(of:100017954)* to the *boss*-method, thus being able to ask *Who is the boss/chief of Microsoft?*. It is important to mention that it can not be assumed that all method names are so 'pure', so that in case the method name does not match a verb or noun we have subcategorization information for, we first substitute punctuation symbols from the method name by spaces, i.e. *wine-region* becomes *wine region* and we consult the subcategorization lexicon again. In case this also doesn't produce any result, we then take the word at the end of the expression, *wine* in our case and consult the lexicon again.

Finally, we derive the entities and np's from membership/subclass statements such as *microsoft:company* or *company::organization* thus yielding the elementary trees depicted in Figure 2.

3 Evaluation

We conducted a small experiment to assess the quality of our lexicon generation component. For this purpose, we selected 5 rather general ontologies from the

Ontology	#Properties	Domain + Range	Non-Composite	Correct	% Correct
Beer	9	4	3	2	66.67%
Wines	10	10	9	6	44.44%
Personal	25	10	8	6	75%
General	28	17	6	6	100%
University	27	1	2	1	50%
Total	99	42	28	21	75%

Fig. 3. Results of the Lexicon Generation Component Evaluation

DAML Ontology Library³ about the following topics: *beer*⁴, *wine*⁵, *general information about organizations*⁶, *personal information*⁷ and *university activities*⁸. From all these ontologies we took the properties (binary relations) and tested if we could find an appropriate subcategorization frame with our lexicon generation component from which to generate appropriate elementary trees. Table 3 gives the following figures: (1) the ontology in question, (2) the number of DAML properties in the ontology, (3) the number of properties in the ontology with a *domain* and a *range* specified, (4) the number of properties from (3) with a non-composite name, (5) the number of the relations from (4) for which our method found a correct subcategorization frame and (6) the percentage of correct answers for the properties in (4). The results show that for 3/4 of the properties in (4) we get correct subcategorization frames. Of course, these results are limited as we have only tested binary properties. It is also clear that the approach needs to be extended to also handle properties with composite names to achieve better results. Nevertheless, these first results are encouraging.

4 Conclusion and Outlook

We have presented ORAKEL, a natural language interface to an F-Logic knowledge base which makes use of a compositional semantics approach to translate wh-questions into F-Logic queries. For this purpose, we have assumed the declarative formulation of LTAG in [13] as well as the extension in [4] as underlying syntactic theory. The approach has been implemented in Java as a parser operationalizing the calculus in [13] and taking into account ontological information as described in [4]. Further, we have presented in detail and evaluated the system's lexicon generation component.

³ <http://www.daml.org/ontologies/>

⁴ <http://www.cs.umd.edu/projects/plus/DAML/onts/beer1.0.daml>

⁵ <http://ontolingua.stanford.edu/doc/chimaera/ontologies/wines.daml>

⁶ <http://www.cs.umd.edu/projects/plus/DAML/onts/general1.0.daml>

⁷ <http://www.cs.umd.edu/projects/plus/DAML/onts/personal1.0.daml>

⁸ <http://www.cs.umd.edu/projects/plus/DAML/onts/univ1.0.daml>

We are currently devising several experiments to evaluate our system. First we intend to acquire typical wh-questions in a Wizard-of-Oz style experiment, manually translate them into F-Logic queries and then evaluate our system in terms of precision/recall with regard to them. This will show how good our system is in dealing with typical questions. Second, we intend to evaluate the usability of the system by comparing the number of successful/failed questions, elapsed time etc. between people asking wh-questions and people directly formulating logical queries to the Ontobroker system.

References

1. I. Androutsopoulos, G.D. Ritchie, and P. Thanisch. Natural language interfaces to databases—an introduction. *Journal of Language Engineering*, 1(1):29–81, 1995.
2. B.K. Boguraev and K. Sparck Jones. How to drive a database front end to databases with evaluative feedback. In *Proceedings of the Conference on Applied Natural Language Processing*, 1983.
3. P. Cimiano. Translating wh-questions into f-logic queries. In R. Bernardi and M. Moortgat, editors, *Proceedings of the CoLogNET-ElsNET Workshop on Questions and Answers: Theoretical and Applied Perspectives*, pages 130–137, 2003.
4. P. Cimiano and U. Reyle. Ontology-based semantic construction, underspecification and disambiguation. In *Proceedings of the Prospects and Advances in the Syntax-Semantic Interface Workshop*, 2003.
5. E.F. Codd. Seven steps to RENDEZVOUS with the casual user. In J. Kimbie and K. Koffeman, editors, *Data Base Management*. North-Holland publishers, 1974.
6. A. Copestake and K. Sparck Jones. Natural language interfaces to databases. *Knowledge Engineering Review*, 1989. Special Issue in the Applications of Natural Language Processing Techniques.
7. S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In *Database Semantics: Semantic Issues in Multimedia Systems*, pages 351–369. Kluwer, 1999.
8. C. Fellbaum. *WordNet, an electronic lexical database*. MIT Press, 1998.
9. B.J. Grosz, D.E. Appelt, P.A. Martin, and F.C.N. Pereira. Team: An experiment in the design of transportable natural language interfaces. *Artificial Intelligence*, 32:173–243, 1987.
10. EW Hinrichs. The syntax and semantic of the janus semantic interpretation language. Technical Report 6652, BBN Laboratories, 1987.
11. A.K. Joshi and Y. Schabes. Tree-adjoining grammars. In *Handbook of Formal Languages*, volume 3, pages 69–124. Springer, 1997.
12. M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the Association for Computing Machinery*, May 1995.
13. Reinhard Muskens. Talking about trees and truth-conditions. *Journal of Logic, Language and Information*, 10(4):417–455, 2001.
14. Philip Resnik. Selectional preference and sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, 1997.
15. Helmut Schmid. Lopar: Design and implementation. In *Arbeitspapiere des Sonderforschungsbereiches 340*, number 149. 2000.

Accessing an Information System by Chatting

Bayan Abu Shawar and Eric Atwell

School of Computing, University of Leeds, LS2 9JT, Leeds, UK

{bshawar, eric}@comp.leeds.ac.uk

<http://www.comp.leeds.ac.uk/eric/demo.html>

Abstract. In this paper, we describe a new way to access information by “chatting” to an information source. This involves a chatbot, a program that emulates human conversation; the chatbot must be trainable with a text, to accept input and match it against the text to generate replies in the conversation. We have developed a Machine Learning approach to retrain the ALICE chatbot with a transcript of human dialogue, and used this to develop a range of chatbots conversing in different styles and languages. We adapted this chatbot-training program to the Qur'an, to allow users to learn from the Qur'an in a conversational information-access style. The process and results are illustrated in this paper.

1 Introduction

The widespread use of Internet and web pages necessitate an easy way to remotely access and search a large information system. Information extraction and information retrieval are about finding answers to specific questions. Information retrieval (IR) systems are concerned with retrieving a relevant subset of documents from a large set according to some query based on key word searching; information extraction is the process of extracting specific pieces of data from documents to fill a list of slots in a predefined templates.

However, sometimes we have less specific information access requirements. When we go to a conference and talk to research colleagues, we tend not to ask questions eliciting specific pointers to sources; rather, we chat more generally and we hope to get a bigger picture of general research ideas and directions, which may in turn inspire us to new ideas. An analogous model for information access software is a system which allows the information-seeker to “chat” with an online information source, to interact in natural language, and receive responses drawn from the online sources; the responses need not be direct “answers” to input “queries/questions” in the traditional sense, but should relate to my input sentences. The conversation is not a series of specific question-answering couplets, but a looser interaction, which should leave the user with an overall sense of the system's perspective and ideas on the topics discussed. As a concrete example, if we want to learn about the ideas in the Qur'an, the holy book of Islam, then a traditional IR/IE system is not what we need. We may use IE to extract a series of specific formal facts, but to get an overview or broader feel for what the Qur'an teaches about broad topics, we need to talk around these topics with an expert on the Qur'an, or a conversational system which knows about the Qur'an.

In this paper we present a new tool to access an information system using chatting. Section 2 outlines the ALICE chatbot system, the AIML language used within it, and the machine learning techniques we used to learn Categories from a training corpus. In section 3 we describe the system architecture we implemented using the Qur'an corpus. We were able to generate a version of ALICE to speak like the Qur'an; this version and samples of chatting are discussed in section 4. Section 5 discusses the evaluation and the usefulness of such a tool. Section 6 presents some conclusions.

2 ALICE and Machine Learning Chatbots

In building natural language processing systems, most NLP researchers focus on modeling linguistic theory. [1] proposes an alternative behavioural approach: "Rather than attacking natural language interactions as a linguistic problem, we attacked it as a behavioural problem... The essential question is no longer "How does language work?" but rather, "What do people say?" ALICE [2], [3] is a chatbot system that implements human dialogues without deep linguistic analysis. The Alice architecture is composed of two parts: the chatbot engine and the language knowledge model. The chatbot engine is a program written in different languages, the one we used is written in Java, and is used to accept user input, search through the knowledge model and return the most appropriate answer.

The language knowledge model is based on manual authoring, which means the information model is hand crafted and it needs months to create a typical knowledge base. This knowledge is written using AIML (Artificial Intelligent Mark up Language), a version of XML, to represent the patterns and templates underlying these dialogues. The basic units of AIML objects are categories. Each category is a rule for matching an input and converting to an output, and consists of a pattern, which represents the user input, and a template, which implies the ALICE robot answer. The AIML pattern is simple, consisting only of words, spaces, and the wildcard symbols _ and *. The words may consist of letters and numerals, but no other characters. Words are separated by a single space, and the wildcard characters function like words. The pattern language is case invariant.

Since the primary goal of chatbots is to mimic real human conversations, we developed a Java program that learns from dialogue corpora to generate AIML files in order to modify ALICE to behave like the corpus. Chatbots have so far been restricted to the linguistic knowledge that is "hand-coded" in their files. The chatbot must be trainable with a text, to accept input and match it against the text to generate replies in the conversation. We have worked with the ALICE open-source chatbot initiative: in the ALICE architecture, the "chatbot engine" and the "language knowledge model" are clearly separated, so that alternative language knowledge models can be plugged and played. We have techniques for developing conversational systems, or chatbots, to chat around a specific topic: the techniques involve Machine Learning from a training corpus of dialogue transcripts, so the resulting chatbot chats in the style of the training corpus [4], [5], [6]. For example, we have a range of different chatbots trained to chat like London teenagers, Afrikaans-speaking South Africans, etc by using text transcriptions of conversations by members of these groups. User input is

effectively used to search the training corpus for a nearest match, and the corresponding reply is output.

If, instead of a using a dialogue transcript, we use written text as a training corpus, then the resulting chatbot should chat in the style of the source text. There are some additional complications; for example, the text is not a series of “turns” so the machine-learning program must decide how to divide the text into utterance-like chunks. Once these problems are solved, we should have a generic information access chatbot, which allows general, fuzzy information access by chatting to a source.

3 System Architecture of the Qur'an Chatbot

The Qur'an is the holy book of Islam, written in the classical Arabic form. The Qur'an consists of 114 sooras, which could be considered as chapters, grouped into 30 parts. Each soora consists of more than one ayyaa (sections). These ayyaas are sorted, and must be shown in the same sequence. In this chatbot version, we used the English/Arabic parallel corpora. The input is in English, and the output is ayyaas extracted from Qur'an in both English and Arabic. The program is composed of three subprograms. The first subprogram is used to build the frequency list from the local database. The second one is used to originate patterns and templates. And the last is used to rearrange the patterns and templates and generates the AIML files. In the following subsections we describe the three subprograms in detail.

3.1 Program (1): Creating the Frequency List

The original English text format looks like:

```
THE UNITY, SINCERITY, ONENESS OF GOD, CHAPTER NO. 112
With the Name of Allah, the Merciful Benefactor, The Merciful Redeemer
112.001 Say: He is Allah, the One and Only;
112.002 Allah, the Eternal, Absolute;
```

The first line represents the soora title and number. The second line is the opening statement, which appears in every soora except soora number (9). The Following lines hold the ayyaas, where each ayyaa has a unique number represented the soora number and the ayyaa number. After removing the numbers, a tokenization process is activated to calculate word frequencies.

3.2 Program (2): Originating Patterns and Templates

Here we used the same program of generating other dialogue chatbots with some modifications. The Arabic corpus was added during this phase. Because of the huge size of the Qur'an text, we split the English/Arabic text into sub texts. Three processes are used within this program as follows:

1. Reading and concatenation process: the program starts by applying three reading processes; reading the English text with its corresponding Arabic one. Reading the English frequency list and inserting the words and its frequencies in least and count lists. Any ayyaa splits over two or more lines are merged together, so each element of the list represents a whole ayyaa.
2. Reiteration and finding the most significant word: since the Qur'an is not a series of "turns" so the machine-learning program must decide how to divide the text into utterance-like chunks. We proposed if an input was an ayyaa then the answer would be the next ayyaa in the same soora. During this process each element of the list except the opening line is repeated to be a pattern in one turn and as a template in the next. The list is now organized such that even indices holding the patterns while the odd ones holding the templates. A tokenization process is activated on each pattern. Then the frequency of each token is extracted from the count list produced in program (1). Finally the least frequent token is extracted.
3. Originating patterns/templates: after finding the least frequent word, the atomic and default categories are originated as follows; the atomic category will have the English pattern, which is the ayyaa, and its English template, which is the next ayyaa. The default category will have the least frequent word(s) as a pattern connected with "*" in different position to match any word, and the template is the ayyaa(s) holding this word(s). During this phase, the English and Arabic soora numbers are replaced by the corresponding soora names. The Arabic template is appended to the English one. At the end, the generated patterns and templates are copied to a file.

3.3 Program (3): The Restructuring Process

Two lists are generated in this process: one for the atomic and another for the default categories. Given a large training corpus like Qur'an, where some ayyaas may be said more than once in the same soora or in different sooras, the file must be restructured to collate these repetitions. At the end the final categories are copied in AIML files.

4 Results

Before retraining ALICE with the generated AIML files, these files must be refashioned to enable ALICE interpreter to recognise the Arabic characters. This implies encoding the files using UTF-8 code. After all, two versions of ALICE were published using the Pandorabot service [8], the first named Qur'an0-30, which handles sooras from 1 to 30, and the second is Qur'an14-114, which handles sooras from 14-114. The program was able to generate 122,469 categories scattered in 45 files. Chatting (1) illustrates the sort of chat a user can have with our Qur'an chatbot.

Chatting 1: chatting with Qur'an14-114

Human: Is it right that our protector is Allah?

Qur'an14-114: {Muhammad (Muhammad)} 11 That is because Allah is the Protector of those who believe, but those who reject Allah have no protector.

11(دِمْ حَ قُرُونُسْ بَلَكْ بَأْنَ اللَّهَ مَوْلَى الَّذِينَ آمَنُوا وَأَنَّ الْكَافِرِينَ لَا مَوْلَى لَهُمْ

{The Morning Star (At-Tariq)} 4 There is no soul but has a protector over it.

4(قُرَاطِلَا قُرُونُسْ إِنْ كُلُّ نَفْسٍ لَمَّا عَلَيْهَا حَافِظَ

Human: Allah, the Eternal, Absolute

Qur'an14-114: {Sincerity (Al-Iklaas)} 3 He begetteth not, nor is He begotten; 3(صَلَخْ إِنْ أَقْرُونُسْ كُلُّمْ يَلَّا وَتَمْ يُولَّا

Human: He begetteth not, nor is He begotten

Qur'an14-114: {Sincerity (Al-Iklaas)} 4 And there is none like unto Him. 4(صَلَخْ إِنْ أَقْرُونُسْ كُلُّمْ يَكْنَى لَهُ شَفَّارَا لَكَنْ

The user types an utterance, such as a question or a statement; the system responds with one or more quotations (sooras and ayyaas) from the Qur'an, which seems appropriate. As this is a chat rather than accessing an information system, the ayyaas found are not simply the result of keyword-lookup; the response-generation mechanism is in fact hidden from the user, who will sometimes get the response "I have no answer for that".

5 Evaluation

Evaluation of this kind of general information access is not easy. As the information accessed is not in terms of specific questions, we cannot count numbers of "hits" in order to compute precision and recall scores. Up to now we evaluated chatting with Qur'an based on user satisfaction. We asked Muslims users to try it. And we asked them to answer the following questions: do you feel you have learnt anything about the teachings of the Qur'an? Do you think this might be a useful information access mechanism? If so, who for, what kinds of users? Some users found the tool unsatisfactory since it does not provide answers to the questions. However using chatting to access an information system, can give the user an overview of the Qur'an contents. It is not necessary that the user will have a correct answer for their request, but at least there is a motivation to engage in a long conversation based in using some of the outputs to know more about the Qur'an. Others found it interesting and useful in the case of a verse being read and the user wants to find out from which soora it came from. This would also benefit those who want to know more about the religion to learn what the Qur'an says in regards to certain circumstances, etc. They consider chatting with Qur'an as a searching engine with some scientific differences.

This tool could be useful to help students reciting Qur'an. All Muslims are taught to recite some of the Qur'an during school. However students usually get bored of the traditional teaching, such as repeating after the teacher or reading from the holy book. Since most students like playing with computers, and chatting with friends, this tool may encourage them to recite certain soora by entering an ayyaa each time. Since it is

a text communication, students must enter the ayyaa to get the next one, and this will improve their writing skills.

In previous research we evaluated our Java program depending on three metrics: dialogue efficiency, dialogue quality, and user satisfaction. From the dialogue efficiency and quality we aim to measure the success of our machine learning techniques. The dialogue efficiency measures the ability of the most significant word to find a match and give an answer. In order to measure the quality of each response, we classified the responses to three types: reasonable, weird but reasonable, or nonsensical. The third aspect is the user satisfaction. We applied this methodology on the Afrikaans dialogues [7], and we plan to apply it to the Qur'an chatbot as well.

6 Conclusions

This paper presents a novel way of accessing information from an online source, by having an informal chat. ALICE is a conversational agent that communicates with users using natural languages. However ALICE and most chatbot systems are restricted to the knowledge that is hand-coded in their files. We have developed a java program to read a text from a corpus and convert it to the AIML format used by ALICE. We selected the Qur'an to illustrate how to convert a written text to the AIML format to retrain ALICE, and how to adapt ALICE to learn from a text which is not a dialogue transcript. The Qur'an is the most widely known Arabic source text, used by all Muslims all over the world. It may be used as a search tool for ayaas that hold same words but have different connotations, so learners of the Qur'an can extract different meaning from the context, it may be good to know the soora name of a certain verse. Students could use it as a new method to recite the Qur'an.

References

1. Whalen, T.: Computational Behaviourism Applied to Natural Language, [online], <http://debra.dgrc.crc.ca/chat/chat.theory.html> (1996)
2. ALICE. A.L.I.C.E AI Foundation , <http://www.Alicebot.org/> (2002)
3. Abu Shawar, B. and Atwell, E.: A Comparison Between Alice and Elizabeth Chatbot Systems. School of Computing research report 2002.19, University of Leeds (2002)
4. Abu Shawar, B. and Atwell E.: Machine Learning from Dialogue Corpora to Generate Chatbots. Expert Update Journal, Vol. 6. No 3 (2003) 25-29.
5. Abu Shawar, B. and Atwell, E.: Using Dialogue Corpora to Train a Chatbot in: Archer, D, Rayson, P, Wilson, A & McEnery, T (eds.) Proceedings of CL2003: International Conference on Corpus Linguistics, Lancaster University (2003) 681-690.
6. Abu Shawar, B. and Atwell, E.: Using the Corpus of Spoken Afrikaans to Generate an Afrikaans Chatbot. to appear in: SALALS Journal of Southern African Linguistics and Applied Language Studies, (2003).
7. Abu Shawar, B. and Atwell, E.: Evaluation of Chatbot Information System. To appear in MCSEA'I04 proceedings (2004).
8. Pandorabot: Pandorabot chatbot hosting service, <http://www.pandorabots.com/pandora> (2003)

Ontology-Based Question Answering in a Federation of University Sites: The MOSES Case Study

P. Atzeni³, R. Basili¹, D.H. Hansen², P. Missier³,
Patrizia Paggio², Maria Teresa Pazienza¹, and Fabio Massimo Zanzotto¹

¹Dip. di Informatica Sistemi e Produzione, University of Rome “Tor Vergata”
`{basili, pazienza, zanzotto}@info.uniroma2.it`

²Centre for Language Technology, University of Copenhagen
`{patrizia, dorte}@cst.dk`

³Dip di Informatica e Automazione, Università Roma Tre
`atzeni@dia.uniroma3.it, pmissier@acm.org`

Abstract. This paper deals with a new approach to ontology-based QA in which users ask questions in natural language to knowledge bases of facts extracted from a federation of Web sites and organised in topic map repositories. Our approach is being investigated in the context of the EU project MOSES.

1 Introduction

This paper deals with a new approach to ontology-based QA in which users ask questions in natural language to knowledge bases of facts extracted from a federation of Web sites and organised in topic map repositories [7]. Our approach is being investigated in the EU project MOSES¹, with the explicit objective of developing an ontology-based methodology to search, create, maintain and adapt semantically structured Web contents according to the vision of the Semantic Web. MOSES is taking advantage of expertise coming from several fields: software agent technology, NLP, text mining and data management. The test-bed chosen in the project is related to the development of an ontology-based knowledge management system and an ontology-based search engine that will both accept questions and produce answers in natural language for the Web sites of two European universities.

Only some of the aspects of the MOSES methodology are presented. Section 2 describes the requirements of the two user groups and proposes a classification of the questions the system is to support. Section 3 discusses the way in which the project intends to comply with the semantic Web vision, and raises the question of ontology mapping in a multilingual environment. Section 4 describes how question analysis is performed, focusing in particular on the interplay between NLP techniques and on-

¹ MOSES is a cooperative project under the 5th Framework Programme. The project partners are FINSA Consulting, MONDECA, Centre for Language Technology, University of Copenhagen, University of Roma Tre, University of Roma Tor Vergata and ParaBotS.

tological knowledge. It also proposes an approach to the analysis of questions addressed to a federation of sites (federated questions) that relies on ontology mapping. Section 5 is a brief conclusion.

2 Requirements and Questions Classification Framework

In the context of the MOSES project, user groups from an Italian and a Danish University were initially asked to identify a common testbed for benchmarking the QA system under development, assuming their respective University web sites as the target information repositories. They were asked to provide both a conceptualisation of their respective domains, expressed using different ontologies, and a collection of reference questions regarding the web site contents, that would serve both as requirements for the design of the QA system, and as testbed to benchmark its effectiveness. The two resulting ontologies exhibit a partial overlap that makes mapping between them feasible. Similar questions were chosen for each of the sites, according to the following requirements.

First, for each question it should be possible to estimate the expected effort for obtaining an answer without using MOSES, so that a benchmark for effectiveness can be established. Second, each node should accept questions and return answers expressed in the native language of that node. Finally, some of the questions should require a search into multiple content repositories (in our case, the two University test sites). These are called *federated questions*. For instance, asking for all active research on a given topic Europe-wide should involve all the University sites that may offer relevant information. Two main issues arise from this requirement. First, there is an issue of multi-linguality, because questions written in a language may propagate to sites that do not understand that language, and that provide answers in their own language. Second, a common protocol for question and answer exchange across nodes must be in place. The former issue is briefly touched upon in later sections, while the second is beyond the scope of this paper.

The collection of questions resulting from the users' effort, each described by metadata regarding both its structure and content, is proving effective in testing a number of key features derived from the requirements in the prototype system, including: (1) building local ontologies by specialising and extending existing off-the-shelf ontologies, as discussed in Section 3.1; (2) creating a mapping between different ontologies defining a similar semantic domain; (3) providing ontology-based natural language question analysis with multi-lingual support.

The structural classification framework for questions is key to making sure that the test bed covers all useful question types. Therefore, we devote the rest of this section to describing the framework in some detail.

The structure of a question can be classified according to various criteria that relate to specific issues in answering it. The first criterion is the structure of the expected answer. The simplest formats for a response are either structured data types, like relational tuples, or unstructured documents. In addition to these types, however, an ontology-based engine would also be able to return concepts (eg "professor") as well

as concept instances (“prof. John Doe”). This dimension has been investigated in detail in a linguistic classification of the syntactic and semantic features of the questions on which the linguistic analysis is based [11].

The second criterion concerns locality, i.e., whether the question is federated or local to a single site. The third criterion concerns how the source information required to answer the query is organised in the sites. We have three major cases:

- *Information directly provided by the site as a whole:* the information is already on a page (or a set of pages, if we are looking for a set of answers).
- *Information available in the sites and correlated but not aggregated:* the information of interest is spread over a set of pages, related by means of links to be followed to reach them.
- *Information available in the site in an unrelated way:* the information is spread over pages that are not explicitly correlated (for example, the correlation could be established by means of common values that appear in the pages but are not supported by links).

Finally, we classify questions according to how the answer can be obtained without using MOSES. Possible answering procedures are *explicit service*, for example if the site offers an address book or a search facility for finding professors; *one-way navigation*, where the user navigates directly to the answer using link support from the site, without backtracking; and *navigation with backtracking*, where reaching the answer requires trial-and-error navigation (for example, I am looking for a professor, but I do not know which department he/she belongs to, and there is no index).

Here we focus on the distinction between local and federated questions, but all the dimensions have been important to decide how to collect the relevant data for the knowledge bases, and will be relevant parameters for the validation of the system.

3 An Ontology-Based Approach to Question Answering

In MOSES, both questions and domain knowledge are represented by the same ontology description language. The system will be developed in two steps. First, a prototypical implementation is planned to answer questions related to the current “state-of-affairs” of the site to which the question is posed. Then, given a “federation” of sites within the same domain, we will investigate how an ontological approach can support QA across the sites. Answering a question can then be seen as a collaborative task between ontological nodes belonging to the same QA system. Since each node has its own version of the domain ontology, the task of passing a question from node to node may be reduced to a mapping task between (similar) conceptual representations. To make such an approach feasible, a number of difficult problems must still be solved. In this paper, we will provide details on how: to build on existing ontologies and interface between them and language resources; to interpret questions wrt the ontological language; to model the mapping task for federated questions.

3.1 Building on Off-the-Shelf Semantic Web Ontologies

One of the results of the Semantic Web initiative will be the production of many interrelated domain-specific ontologies that provide the formal language for describing the content of Web documents. Since publicly available non-toy ontology examples are already available, the effort of adapting an existing ontology to a specific application is both useful and possible. MOSES has taken the “university ontology” from the DAML-OIL ontology library as a starting point. The classes and relations of this ontology cover in fact, at least at a high level, most of the relevant concepts of the analysed scenarios (i.e. *People/Courses/Research*).

The ontology has been adapted to develop conceptualisations for each of the two national university sub-systems (i.e. Italian and Danish) while providing additional information required to answer the input questions. The Danish ontology contains for example about 200 classes and 50 relations. Instances of the classes are being added by the project’s user groups by downloading them from the respective sites’ databases as well as by manually extracting data from the Web pages. This manually acquired knowledge will be used to develop machine learning algorithms to allow for a semi-manual construction of the domain knowledge.

The first challenge deriving from having two separate ontologies for the same domain is the language. Concepts and relations in the two ontologies are in fact labelled in different languages. This means that a mapping algorithm making use of string similarity measures applied to concept labels will have to work with translation, either directly between the two languages involved, or via a pivot language like English. Another challenge comes from the structural differences: not all the nodes in one ontology are represented also in the other and vice-versa. Finally, domain relations are treated differently in the two ontologies. In the Italian one, all relations are binary in keeping with the original DAML-OIL model, whereas the Danish ontology makes use of n-nary relations in the spirit of the Topic Maps formalism.

3.2 Linguistic Interfaces to Ontologies

Ontologies for the Semantic Web are written in formal languages (OWL, DAML+OIL, SHOE) that are generalisations/restrictions of Description Logics [1]. In these formal languages, to each concept class and relation corresponds one label. As the label has the only purpose of highlighting the concept to human readers, alternative linguistic expressions are not represented. Rather, this piece of information is recorded in a lexical data base like WordNet. The problem is quite obvious when considering binary relationships. For example, the relation **teacherOf** between the concepts **#FacultyMember** and a **#Course** has the label *Teaches*. It does not mention alternative linguistic expressions like: **#Faculty gives #Course** or **#Faculty delivers #Course**, etc.

For the ontology producers, only one concept or relation name is sufficient. Synonymy is not a relevant phenomenon in ontological representations. In fact, it is considered a possible generator of unnecessary concept name clashes, i.e. concept name

ambiguity. Conceptualisations like the one in table 1 are inherently too weak to define linguistic models for NLP applications. Interpreting questions like:

Who gives/teaches the database class/course this year?

with respect to a university domain ontology means in fact mapping all the questions onto the concepts and relations in Table 2. In developing an ontological QA system, the main problem is then to build what we call the “linguistic interface” to the ontology which consists of all the linguistic expressions used to convey concepts and relationships. To make this attempt viable, we are currently studying methods to automatically relate lexical knowledge bases like WordNet [10] to domain ontologies [2] and to induce syntactic-semantic patterns for relationships [3]. Currently, however, the linguistic interface on which the semantic model of the natural language processing sub-system builds, is being developed manually.

4 Question Analysis

Question analysis is carried out in the MOSES linguistic module associated with each system node. To adhere to the semantic Web approach, MOSES poses no constraints on how the conceptual representation should be produced, nor on the format of the output of each linguistic module. The agent that passes this output to the content matcher (an ontology-based search engine) maps the linguistic representation onto a common MOSES interchange formalism (still in an early development phase). Two independent modules have been developed for Danish and Italian language analysis. They have a similar architecture (both use preprocessing, i.e. POS-tagging and lemmatising, prior to syntactic and semantic analyses), but specific parsers. Whereas the Danish parser, an adapted version of PET [5] produces typed feature structures [6], the Italian one outputs quasi-logical forms. Both representation types have proven adequate to express the desired conceptual content, which is expressed in terms of the classes, instances and relations in the two ontologies. As an example, the Italian analysis module is described below.

4.1 Analysis of Italian Questions

Analysis of Italian questions is carried out by using two different linguistic interpretation levels. The syntactic interpretation is built by a general purpose robust syntactic analyser, i.e. Chaos [4]. This will produce a Question Quasi-Logical Form (Q-QLF) of an input question based on the extended dependency graph formalism (XDG) introduced in [4]. In this formalism, the syntactic model of the sentence is represented via a planar graph where nodes represent constituents and arcs the relationships between them. Constituents produced are chunks, i.e. kernels of verb phrases (VPK), noun phrases (NPK), prepositional phrases (PPK) and adjectival phrases (ADJK). Relations among the constituents represent their grammatical functions: logical subjects (lsubj), logical objects (lobj), and prepositional modifiers. For example, the Q-

QLF of the question *Chi insegna il corso di Database?* (*Who teaches the database course?*) is shown in Figure 1.

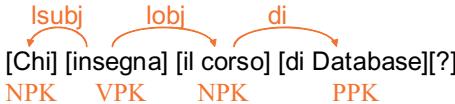


Fig. 1. A Q-QLF within the XDG formalism

Then a robust semantic analyser, namely the Discourse Interpreter from LaSIE [8] is applied. An internal world model has been used to represent the way in which the relevant concepts (i.e. objects) and relationships (i.e. events) are associated with linguistic forms. Under the object node, concepts from the domain concept hierarchy are mapped onto synsets (sets of synonyms) in the linguistic hierarchy EWN (i.e. the EuroWordNet.base concepts). This is to guarantee that linguistic reasoning analysis is made using general linguistic knowledge.

```

TEACH_EVENT ==> teach_course.
teach_course ==> tenere v insegnare v fare.
props(teach_course(E),[(consequence(E,
    [relation(E,teacherOf),r_arg1(E,X),r_arg2(E,Z)] ):-%
        nodeprop(E,lsubj(E,X)), X <- ewn4123(_),/* human_1 */%
        nodeprop(E,lobj(E,Z)), Z <- ewn567704(_))]./* education_1 */)

```

Fig. 2. Example of syntactic-semantic interpretation rule

The association of objects and events with linguistic forms is used in matching rules as shown in Figure 2. The rule expresses the fact that, if any word like *tenere*, *insegnare* or *fare* is encountered in relation with a *human_1* (represented by the base concept *ewn4123*) and the word *education_1* (*ewn567704*), the relation *teacherOf* can be induced. The analysis resulting for the sentence *Chi insegna il corso di Database* is then:

```

focus(e2), relation(e1,teacherOf),
r_arg1(e1, person_dch(e2)),r_arg2(e1,course_dch(e3)),relation(e4,hasSubject),
r_arg1(e4, course_dch(e3)),r_arg2(e4,topic_dch("Database"))).

```

This means that the user is interested in a person, the entity *e2* of the class *person_dch*, that is in a relation *teacherOf* with the entity *e4* (instance of the class *course_dch*), that is in turn related by *hasSubject* with the topic (i.e. *topic_dch*) "Database". This result can be passed on to the content matcher.

4.2 Handling Federated Questions

Now we want to extend this approach to question analysis in order to manage federated questions. A possible solution would be sending the natural language question to several nodes and let each node interpret it against its own domain knowledge. This is unfeasible in a multilingual environment. The solution we are investigating is based

instead on ontology mapping. Let us consider the case of a student questioning not only the Danish but also the Italian site (by selecting specific modalities for entering questions): *Hvem er lektor i fransk?* (*Who is associate professor of French?*)

As the question is in Danish, it has to be analysed by the Danish analysis component, which will produce a semantic interpretation roughly equivalent to the following term: *all(x) (lektor(x) & CourseOffer(x,y) & Course(y) & Name(y, French))*. Since all concepts and relations come from the Danish ontology, it is not a problem to query the Danish knowledge base for all relevant examples. In order to query the Italian knowledge base, however, equivalent concepts and relations must be substituted for those in the “Danish” interpretation. The corresponding Italian representation is: *all(x) (ProfessoreAssociato(x) & TeacherOf(x,y) & Course(y) & Subject(y, French))*

The first problem is establishing a correspondence between ‘lektor’ and ‘ProfessoreAssociato’, which, however, are not structurally equivalent. As suggested in [12,9], equivalence relations must be established by considering *is-a* structures and lexical concept labels together. In the example under discussion, an initial equivalence can be posited between the top nodes of the two ontology fragments, since they both refer explicitly to the original DAML+OIL ontology via a *sameAs* relation. However, none of the concept labels under ‘Faculty’ in the Italian ontology are acceptable translations of ‘Lektor’, nor do any of the nodes refer to common nodes in a common reference ontology. Thus, the matching algorithm must search further down for an equivalent concept by considering possible translations of concept labels and testing the relations that equivalence candidates participate in. Thus, distance from a common starting node, lexical equivalence and occurrence in similar relations are all constraints to be considered.

The same problem of finding a correct mapping applies to relations. In this case, we must be able to discover that *CourseOffer* and *TeacherOf* represent the same relation. For instance we can rely on the fact that they have both two roles, and the concepts filling these roles, Faculty and Course (or rather the Danish and Italian equivalent concepts) correspond. Discovering similarities between relations, however, may be a much more complex task than shown in this example. In general, it assumes the ability to map between concepts.

5 Conclusion

Our focus in this paper has been, in the context of ontology-based QA, to discuss how to interface between ontology and linguistic resources on the one hand, and ontology and natural language questions on the other while remaining within a unique framework. We have described how multilingual input questions are analysed and classified to define the scope of the QA system, and explained the way in which NLP techniques interact with the ontology to produce semantic representations of the questions to be fed to the system’s content matcher. An interesting requirement to the framework we are developing is that it must deal with a multilingual environment. This in turn means supporting questions to federation of sites organised around local ontolo-

gies. It is shown in the paper that this issue can be addressed in terms of ontology mapping, and that mapping algorithms can benefit from looking not only at structural equivalence, but also at similarity of concept labels.

References

1. Baader, F., D. Calvanese, D. McGuinness, D. Nardi, P.F. Patel-Schneider, eds. (2003) *The Description Logics Handbook: Theory, Implementation, and Applications*, Cambridge University Press.
2. Basili, Roberto, Michele Vindigni, Fabio Massimo Zanzotto (2003a) *Integrating ontological and linguistic knowledge for Conceptual Information Extraction*, Web Intelligence Conference, Halifax, Canada, September 2003.
3. Basili, Roberto, Maria Teresa Pazienza, and Fabio Massimo Zanzotto (2003b) *Exploiting the feature vector model for learning linguistic representations of relational concepts* Workshop on Adaptive Text Extraction and Mining (ATEM 2003) held in conjunction with Europena Conference on Machine Learning (ECML 2003) Cavtat (Croatia), September 2003.
4. Basili, Roberto and Fabio Massimo Zanzotto (2002) *Parsing Engineering and Empirical Robustness* Journal of Natural Language Engineering 8/2-3 June 2002.
5. Callmeier, Ulrich (2000) PET – a platform for experimentation with efficient HPSG processing techniques. In Flickinger, D., Oepen, S., Tsujii, J. and Uszkoreit, H. (eds.) *Natural Language Engineering. Special Issue on Efficient Processing with HPSG*. Vol. 6, Part 1, March 2000, 99–107.
6. Copestake, Ann (2002) *Implementing Typed Feature Structure Grammars*. CSLI Publications. Stanford University.
7. Garshol, Lars Marius (2003) Living with Topic Maps and RDF. Technical report. <http://www.ontopia.net/topicmaps/materials/tmrdf.html>.
8. Humphreys, K., R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks (1998) *University of sheffield: Description of the LASIE-II system as used for MUC-7*. In Proceedings of the Seventh Message Understanding Conferences (MUC-7). Morgan Kaufman, 1998.
9. Meadche, Alexander and Steffen Staab (2001) *Comparing Ontologies-Similarity Measures and Comparison Study*, Internal Report No. 408, Institute AIFB, University of Karlsruhe, Germany, 2001.
10. Miller, George A. (1995) WordNet: A lexical database for English. Communications of the ACM, 38(11):39–41, 1995.
11. Paggio, P., D. H. Hansen, M. T. Pazienza, R. Benigni and F. M. Zanzotto (2004) Ontology-based question analysis in a multilingual environment: the MOSES project. *Proceedings of OntoLex 2004*. Lisbon, Portugal, May 2004, pp.1-8.
12. Pazienza, Maria Teresa and Michele Vindigni (2003) *Agent-based Ontological Mediation in IE systems* in M.T. Pazienza ed. *Information Extraction in the Web Era*, LNAI 2700, Springer Berlin 2003.

Semantic Tagging and Chunk-Parsing in Dynamic Modeling

Günther Fliedl¹, Christian Kop¹, Heinrich C. Mayr¹, Christian Winkler²,
Georg Weber¹, and Alexander Salbrechter¹

¹Institute of Business Informatics and Application systems,

²Institute of Linguistics and Computational Linguistics
Klagenfurt University, Austria

Abstract. The paper outlines a multilevel tagging approach to the linguistic analysis of requirements texts. It is shown that extended semantic tagging including chunk-parsing of noun-groups and prepositional groups enables to identify structural items which can be mapped to the conceptual notions for dynamic modeling in KCPM, namely *actor*, *operation-type* and *condition*.

1 Introduction

The design of information systems in general and dynamic modeling in particular usually start with natural language requirement texts that are elicited via dialogues between users and system analysts [3], [4], [5]. As has been shown in previous papers (e.g. [6], [7], [8], [10]), such requirements texts may be analyzed by means of deep parsing as long as they comply with some standardization rules of a given Universe of Discourse (UoD). When describing dynamic models, however, requirement texts mostly contain conditions, i.e., sentences of the type *wenn-dann* (if-then), which often oppose to the methods of deep parsing. We, therefore, suggest for such cases an approach which, in a first step, combines semantic tagging and chunk parsing, and then exploits the results of this step within a subsequent interpretation phase. The latter results in a first cut predesign schema using the modeling notions of KCPM, the Klagenfurt Conceptual Predesign Model [12], which is used as an Interlingua for a subsequent mapping to arbitrary conceptual models.

The paper outlines the main aspects of semantic tagging and its subsequent analysis steps to detect relevant KCPM notions. Therefore in section 2 the necessary notions of KCPM are. Section 3 describes the functionality of NIBA-TAG, a tool we developed for semantic tagging and chunk parsing. Section 4 is devoted to the subsequent analysis and interpretation of the tagging results in order to receive KCPM notions. The paper is summarized in section 5.

2 KCPM Fundamentals

For a better understanding of what follows, the KCPM notions *actor*, *operation type* and *condition* (pre- and post-condition) are shortly introduced. Other KCPM notions

(e.g. thing type, connection type, cooperation type) and details on them can be found in previous papers [12], [6].

Using the framework of the SAMMOA meta model [11], *operation-types* are used to model functional services that can be called via messages (service calls). As such they may be perceived of as a generalization of the notions use-case, activity, action, method, service etc.. Each operation-type is characterized by references to thing-types, which model the *actors* (acting and calling actors of the respective operation-type) and service parameters. An ‘acting actor’ is the thing-type to which the operation-type has been assigned as a service. It is capable of carrying out instances of the operation-type. A ‘calling actor’ is the thing-type that may initiate an instance of an operation-type (i.e. a process) by calling up the respective service of an acting actor. From an object-oriented point of view, this call happens during the execution of an operation by the calling object. Consider, e.g., the sentence ‘*Der Kunde hebt Geld vom Bankomaten ab*’ (the client draws money from the teller machine). we can state: The *object-type* ‘Client’ has a *communication-relationship* to the *object-type* ‘Teller_machine’ based on its *operation* (a feature) ‘draw_money’, which embodies calls of *operations* like ‘accept_card’ and ‘eject-money’ of ‘Teller_machine’. Consequently, KCPM treats operation-types as services offered by acting thing-types. ‘accept_card’ and ‘eject_money’ would thus be modeled as operation-types of acting thing-type ‘Teller_machine’.

UoD dynamics emerge from (possibly concurrent) activities of (one or more) actors that are performed under certain circumstances (*pre-conditions*) and that create new circumstances (*post-conditions*). This is captured by the KCPM concept of cooperation-type.

3 Linguistic Framework

For the purpose of Requirements Engineering we picked out the concept of tagging, determining whether it is useful. The classical approaches of tagging [1] [2] use standardized tag sets like the Brown/Penn set [13]. The Penn-Treebank tag set has been the one most widely used in computational work. It is a simplified version of the Brown tag set and includes tags like VBP (main verb in present form), VB (verb in the infinitive), VBD (verb in past tense), VBG (past passive participle) etc..

This type of pure standardized tagging (e.g. Treetagger, Brilltagger, Morphy, Q-tag etc) seems to have the following three deficits:

- Tags provide merely categorial-syntactic information,
- ambiguity can be made explicit only in a restricted sense,
- only limited chunking (normally chunking of NPs) is possible.

Our main goal of natural language based requirements engineering is undoubtedly the extraction of semantic concepts out of linguistic structures. A somehow more sophisticated tagging instrument that assigns semantic tags with morphosyntactic relevance is needed. Therefore we developed NIBA-TAG, a tool for tagging and chunking natural verb tags and assigning morphosyntactically relevant semantic tags. NIBA-TAG is a multilevel natural language tagger that integrates the functionality of a word-

stemmer, a morphological parser and a conventional POS-Tagger for German. Based on the NTMS [9] it uses a fine granulated set of semantically motivated verb class tags, which can be seen as triggers for predicate argument structures (PAS) including semantic roles like **AG**(Agens), **EXP**(Experiencer), **TH**(Patiens/Objective), **CAUSE**, **GO**(Goal), **SO**(Source), **LOC**(Location) and **INSTR**(Instrument) for verb classification.

Consider the following input text:

- (1) *Der Auftrag trifft ein. (The order comes in.)*
- (2) *Die Auftragsabteilung prüft jeden Artikel des Auftrags. (The ordering department checks each item of the order.)*

NIBA-TAG will generate the following output for the first sentence.

```
<sentence position="0" id="0">
  <n3 position="0">
    <spz0 referTo="der" position="0" id="0" lowerCa-
      se="der">Der
    </spz0>
    <n0 referTo="Auftrag" lastposition="1" positi-
      on="1" id="1" lowerCase="auftrag">Auftrag
    </n0>
  </n3>
  <v0 referTo="eintreffen" partikel="ein" position="1"
    id="2" lowerCase="trifft" verbclass="eV">trifft
  </v0>
  <pt0 lastposition="1" position="2" id="3" lower-
    Case="ein">ein
  </pt0>
</sentence>
```

The Part-of-speech tags in the XML-output generated by NIBA-TAG relate words to categories, based on how they can be combined to form sentences. For example, articles can combine with nouns, but not with verbs. Thus the Noun phrase tag (N3 tag) `<n3>` represents this category. As outlined above, the NIBA-TAG tags also information about the semantic content of a word. Consider, e.g., `eV` which is the value of verbclass argument in the tag `<v0>`. It stands for the concept [*-agentive person + terminative process*]. On the other hand a value `tVag2` in that argument decodes [*agentive person + intentional activity implying bivalence*]. This would be the case for the verb “*prüft*”(checks) in the second sentence.

4 Interpretation of the Tagging Results

Using the results of the semantic tagging and chunk parsing of NIBA-TAG, interpretation is done in two steps:

1. The fundamental relationships between the respective phrases have to be found out based on the verb category, PAS, sentence mode (active voice, passive voice), type of clause (main clause) and the identification of noun phrases (N3 tags).
2. After that, these relationships are interpreted using the KCPM dynamic model.

(1) *Der Auftrag trifft ein.*

In the example above the verb is an ergative verb (verb class = ‘eV’). According to their PAS, ergative verbs have an internal argument which has to be found now. On the base of the tagging output, there are two alternatives where to search.

- a) The first N3 tag before the verb - or it’s auxiliary - is its argument (*Der Auftrag – the order*).
- b) If the sentence does not start with a N3 tag, e.g. in the case of a topicalization of subordinate clause or a preposition, then the argument is the N3 tag directly after the verb – or it’s auxiliary.

With respect to this heuristic the following fundamental argument relationship between the noun and the verb can be found.

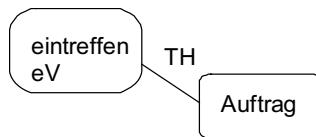


Fig. 1. Relationship between verb (*eintreffen*) and noun (*Auftrag*)

Ergative verbs are candidates for conditions [6],[7]. A KCPM condition consists of the thing-type involved in the condition and the condition itself, which can be fulfilled or not. Since ergative verbs imply only one argument with the semantic role TH (see fig. 1), the corresponding noun is the identified thing-type involved. In the example “*Der Auftrag trifft ein*”, *Auftrag* thus is the involved thing-type and *trifft_ein* the condition, which can be fulfilled or not. For other verbs topicalisation of an argument (subject or object) indicates which thing-type is preferred for the involved thing type of a condition.

(2) *Die Auftragsabteilung prüft jeden Artikel des Auftrags.*

The tagger identifies *prüfen* (to check) as the finite main verb of the phrase, assigning to it the semantic category *tVag2*. *prüfen* has thus to be interpreted as a bivalent agent verb. The semantic tag implies two arguments (AGENT and THEMA). The noun phrase (in our model N3) located before the verb of a main clause is by default identified as an argument.

Yet, since in German either the subject or the object may be placed before the verb, the argument type cannot be identified (e.g. *Jeden Artikel des Auftrags prüft die Auftragsabteilung*). Due to the fact that the Quantifier *jeden* (each, every) clearly encodes the accusative, the accusative object is the internal argument.

The third N3-tag cannot be interpreted as a (separate) argument of the verb, since the verb class involved restricts the number of arguments to 2 (A checks B). Furthermore, the definite article *des* (of the/s) encodes the relation (inclusion) between *Artikel* (item) and *Auftrag* (order).

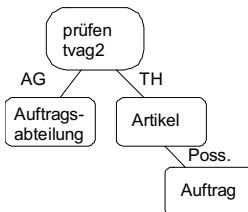


Fig. 2. Relationship between verb (*prüfen*) and nouns (*Auftragsabteilung*, *Artikel*, *Auftrag*)

Priüfen (to check) is a bivalent agent verb. Agentive verbs provide information regarding operation-types. *Auftragsabteilung* in this relationship carries the role AG, *Artikel* the role TH. With respect to their semantic roles, *Auftragsabteilung* and *Artikel* are thing-types. According to the SAMMOA model, this expression can be perceived of as a relationship of communication. Because of the AGENT-role (carrier of action), *Auftragsabteilung* in this relationship is an actor. Whether it is an acting or a calling actor can not be decided independently from context, though.

5 Conclusion

The interpretation of morphosyntactic and semantic tags allows a direct conceptualization of modeling relevant information. The core concepts of dynamic modeling in KCPM (*actor*, *operation-type*, *pre-condition*, *post-condition*) have been put on a linguistic fundament. Moreover, employing an extended tagging system has provided a respectable set of key tags referring to verb classes, which are used for the classification of about 16.000 structured Standard German verb entries.

References

- Brants, T.: TnT - A Statistical Part-of-Speech Tagger. In: Proc. of the 6th Applied Natural Language Processing Conference ANLP-2000. Seattle, pp. 224–231
- Brill, E.: A simple rule-based part of speech tagger. In: Proc. of the 3rd Conference on Applied Natural Language Processing, ACL, 1992
- Buchholz, E., Cyriaks, H., Düsterhöft, A., Mehlan, H., Thalheim, B.: Applying a Natural Language Dialogue Tool for Designing Databases. In: Proc. Int. Workshop on Applications of Natural Language to Databases (NLDB'95), France, 1995
- Burg, J.F.M.: Linguistic Instruments in Requirements Engineering. IOS Press, Amsterdam u.a., 1997
- Ceri, S. (ed.): Methodology and Tools for Database Design. North Holland Publ. Comp., 1983
- Fliedl, G., Kop, Ch., Mayerthaler, W., Mayr, H.C., Winkler, Ch.: Guidelines for NL-Based Requirements Specifications in NIBA. In (Bouzeghoub, M., Kedad, Z., Metais E. eds.): Proc. 5th Int. Conf. on Applications of Natural Language to Information Systems (NLDB'2000). Lecture Notes in Computer Science (LNCS 1959), Springer Verlag, 2000, pp. 251–264

7. Fliedl, G., Kop, Ch., Mayerthaler, W., Mayr, H. C., Winkler, Ch.: Linguistic Aspects of Modeling Dynamics. In: Proc. 2nd Int. Workshop on Natural Language and Information Systems (NLIS 2000), Greenwich, UK, 2000, pp. 83–90
8. Fliedl, G., Kop, Ch., Mayerthaler, W., Mayr, H. C., Winkler, Ch.: Linguistically based requirements engineering - The NIBA project. In: Data & Knowledge Engineering, Vol. 35, 2000, pp. 111-120
9. Fliedl, G.: Natürlichkeitstheoretische Morphosyntax, Aspekte der Theorie und Implementierung. Gunter Narr Verlag, Tübingen, 1999
10. Fliedl, G., Mayerthaler, W., Winkler, Ch.: The NT(M)S Parser: An Efficient Computational Linguistic Tool. In: Proc. 1st Int. Workshop on Computer Science and Information Technologies, Moscow, 1999, pp. 125-128
11. Hesse, W., Mayr, H.C.: Highlights of the SAMMOA Framework for Object Oriented Application Modelling, In (Quirchmayr, G., Schweigofer, E., Bench-Capon, J.M. eds.): Proc. 9th Int. Conf. of Database and Expert Systems Applications (DEXA'98), Lecture Notes in Computer Science, Springer Verlag, 1998, pp.353-373
12. Kop, C., Mayr, H.C.: Conceptual Predesign - Bridging the Gap between Requirements and Conceptual Design. In: Proc. 3rd Int. Conf. on Requirements Engineering (ICRE'98), Colorado Springs, April 1998
13. Marcus, M., Santorini, B. and Marcienkiewicz, M.: Building a large annotated corpus of English: the Penn Treebank. Computational Linguistics, 1993

Semantic Filtering of Textual Requirements Descriptions

Jorge J. García Flores*

LaLICC
Université de Paris 4
96, bd Raspail,
75006 Paris, France
jorge.gflores@paris4.sorbonne.fr

Abstract. This paper explores the use of semantic filtering techniques for the analysis of large textual requirements descriptions. Our approach makes use of the Contextual Exploration Method to extract, within large textual requirements descriptions, those statements considered as relevant from requirements engineering perspective: concepts relationships, aspecto-temporal organisation, cause and control statements. We investigate to what extent filtering with these criteria can be the base of requirements analysis and validation processing, and what kind of software tools are necessary to support contextual exploration systems for this purpose.

1 Introduction

Many software development projects fail because of a deficient requirements strategy. Requirements engineering (RE) studies the processes, techniques and methods that would allow a system to meet the purpose for which it was intended [16]. Natural language plays an important role in RE. A recent study [14] shows that 73% of the documents available for requirements analysis are written in natural language. Use cases, scenarios, user stories, transcriptions of conversations for requirements elicitation [8] and even rough sketches are examples of textual requirements descriptions (TRD). Natural Language Processing (NLP) provides useful techniques to extract information from this documents, which can reach several hundred of pages in large projects.

The use of linguistic knowledge for natural language requirements processing is not new. In the past, linguistic instruments have been used to extract conceptual schema from NL requirements ([18,7]), to analyse and validate requirements through a conceptual representation ([3,1]) or to improve the linguistic quality of TRD documents ([17,6]). However, few of these approaches take into account the size of the input text. Large TRD documents raise interesting questions for

* PhD research funded by CONACYT (Mexican Government). Thanks to Joël Bourgeys, Marie-Josée Goulet, Jean-Luc Minel, Camille Salinesi and Marie Chagnoux for their feedback on earlier versions of this paper.

NL requirements processing techniques. What should be the processing's scope? Is it useful to process the whole mass of documents, or is it better to limit it to some extent? If there are formal representations involved, how do they deal with such amount of input text? What are the effects of large TRD on the NL processing strategy?

This paper postulates that filtering relevant text fragments according to semantic criteria enhances large TRD processing. Its purpose is to explore the use of a linguistic technique, the Contextual Exploration (CE) Method [5], to extract semantically relevant sentences in order to support requirements analysis and validation. Four semantic viewpoints are considered as relevant for large TRD processing: 1) concepts relationships, 2) aspecto-temporal organisation, 3) control and 4) causality. The paper is organised as follows: section 2 introduces the CE method. Section 3 is devoted to the discussion of how semantic filtering would support requirements analysis and validation. It proposes an architecture for a TRD semantic filtering system. Section 4 presents the conclusions and sketches future work.

2 The Contextual Exploration Method

In the frame of the Cognitive and Applicative Grammar (GAC in French) linguistic theory [4], the CE Method [5] was originally designed to solve lexical and grammatical polysemy problems. The method rests on the principle that domain independent linguistic elements structure text meaning. Its purpose is to access the semantic content of texts in order to extract relevant information according to a certain task.

According to contextual exploration, all signs occurring in a text (the textual context) must be taken into account to determine the semantic value of a sentence. This example illustrates how indeterminacy is solved¹:

- (1) *In spite of all precautions, he was captured the day after*
- (2) *Without all precautions, he was captured the day after*

From the aspecto-temporal point of view, the tense of “was captured” is the linguistic marker of a semantic value that can be NEW-STATE (he was captured) or UNREALIZED (he was not captured). However, the tense itself is not enough to decide which one of both values must be assigned, so the context has to be analysed in order to get more clues. In this case “in spite” and sans “without” are the clues that determine the semantic values of these sentences.

Linguistic markers correspond to plausible hypothesis that must be confirmed by the presence of certain clues. The heuristic mechanism is based in rules. A rule R is defined as follows:

$$R_k = [K, \{I_p, C_p\}, D_k]$$

¹ This example is from G. Guillaume, quoted by Desclés. [5]

Where K is a class of linguistic marker, I a class of clue and, C the research context and D the decision to take [15]. Rules are organised by tasks; each task conveys a particular semantic notion. For instance, the task *Static Relations* [13] specifies a set of 238 rules, 6149 markers and 1777 clues in order to assign a set of 14 semantic values. This task would assign the INGREDIENTE semantic value to the following sentences (taken from a TRD of an insurance system):

- (3) *A joint policy **includes** the joiners age, the joiners gender and his smoker condition.*
- (4) *The agent displays billing mode, effective date, due date, bill number and total premium amount, which **are part of** the policy's billing detail.*
- (5) **For each** mandatory rider, the agent should specify the following values:
 - Face amount
 - Due period
 - Increased Periods

Within the “concepts relationships” viewpoint, the INGREDIENTE semantic value filters ”part-of” relations from a text. In the above example linguistic markers are highlighted in bold. In sentence (5) the linguistic marker “For each” is not strong enough to assign the INGREDIENTE value, but its plausibility may be confirmed by non-lexical clues, like the typographical sign ":" or the presence of a list. Linguistic clues may not only be lexical entries. Punctuation signs and text structural elements, like titles and sections, are allowed in a clue class.

So far, the CE method has been applied to automatic summarisation [15], filtering of causality relations [9], relations between concepts [13], aspecto-temporal relations and to extract information from the web using semantic viewpoints [12]. The CE method present specific aspects that distinguish it from other semantic tagging techniques used in the context of NL requirements processing:

1. It is oriented to large document sets, so it can handle large TRD while taking account of the linguistic context.
2. It has rule-based structure that could allow to define semantic validations beyond the limits of a sentence, and even between requirements that are far away from each other in the requirements document.
3. CE markers and clues are not restricted to linguistic elements, but also typographical and structural ones. Despite the importance of structural elements (titles, subtitles, etc.) in industrial TRD documents, little research has been done on their exploitation by NL requirements processing tools.
4. CE semantic tags are not tight to any software design ”methodology”. The difference between CE an other methodology neutral approaches ([1,2]) is that CE produce an abridged text unit (the filtered text) which could be the input to others NL requirements processing tools.

3 Semantic Filtering of TRD

Semantic filtering aims at improving large TRD processing by drawing together a small number of statements that share a certain semantic viewpoint. However, extraction is made using relevance criteria, and relevance, as it has been pointed out by Minel [15], depend for the most part on the reader's point of view. The following viewpoints are considered as semantically relevant from a RE point of view: I) Concepts relationships, II) Aspecto-temporal organisation, III) Control and IV) Causality.

Each one of this viewpoints conveys a semantic concept that, according to the GAC linguistic model, may structure and organise meaning [4], and each one represents an important aspect in requirements analysis as well. Much effort have been devoted to build conceptual schema from TRD's relations between concepts for requirements analysis ([18,7,1]). The value of extracting event's and processes temporal organisation (the "dynamic" aspect) for NL requirements validation has been remarked by Burg [3]. Control issues, i.e. specifying if actions are "environment controlled" or "machine controlled", are of primary importance in RE [19], and a precise understanding of the causal organisation of actions is necessary to specify the rules that a system must obey [11]. The following example shows different views of one TRD paragraph according to these semantic viewpoints. Relevant sentences are marked in bold. Under-braces indicate semantic values assigned by CE rules according to linguistic markers and clues (underlined)²:

- Relations between concepts viewpoint:

When the start button is pressed, if there is an original in the feed slot, the photocopier makes N copies of it, and places them in the output tray. N is the number currently registering in the count display. If the
(EQUALITY)

start button is pressed while photocopying is in progress, it has no effect. The number N in the count display updates in response to button pressed according to the state table.

- Aspecto-temporal organisation viewpoint:

When the start button is pressed, if there is an original in the feed slot, the photocopier makes N copies of it, and places them in the output tray. N is the number currently registering in the count display. If the start button is pressed while photocopying is in progress, it has no effect. The number N in the count display updates in response to button pressed according to the state table.

- Control viewpoint:

When the start button is pressed, if there is an original in the feed slot, the photocopier makes N copies of it, and

(MACHINE CONTROLLED)

² This example is taken from a requirements document from Kovitz [11]

places them in the output tray. N is the number currently registering in the count display. If **the start button is pressed** while photocopying is in progress, it has no effect. **The number N in the count display updates in response to button pressed according to the state table.**

- Causality viewpoint:

When the start button is pressed, if there is an original in the feed slot, the photocopier makes N copies of it, and places them in the output tray. N is the number currently registering in the count display. If the start button is pressed while photocopying is in progress, it has no effect. The number N in the count display updates in response to button pressed according to the state table.

Every viewpoint would produce, after TRD filtering, a two-folded output: the filtered text and its associated semantic values. Our first hypothesis is that semantic values may allow to find semantic conflicts in large TRD, specially in requirements that are far away from each other in the TRD document. The following is an example taken from a in insurance system TRD:

- (page 60) *When the product's life-cycle is over, the system should trigger a premium-collection event.*
 (page 234) *The system can prevent a premium-collection event but only an agent can cause it.*

In the statement of page 234, filtered by the causality viewpoint, establishes that only an agent can *cause* a premium-collection event, while in page 60's statement (control viewpoint) there are traces of a machine-controlled situation over the premium collection event, where the system *triggers* the event.

The detection of conflicts needs the definition of validation rules based on semantic values. These semantic-value based rules would allow to detect inter-viewpoint conflicts and conflicts inside a viewpoint (for instance, to verify that, in INGREDIENCE relations from the concepts relationships viewpoint, the relation between an element and its parts does never gets reversed). They allow requirements analysis as well (for instance, a high proportion of cause and control statements may be sign of a Jackson's control problem frame [10]).

Our second hypothesis is that the filtered texts can be the input of other NL requirements processing tools that don't support large TRD processing. For instance, the conceptual schema generation tools from CIRCE [1] or COLORX [3] could be used to process the filtered text, regardless of its semantic values.

Figure 1 shows the proposed architecture to supports TRD semantic filtering. The rules configuration tool is intended to support a linguistic configuration phase, where the application's domain glossary is included into the linguistic resources, the CE and semantic-value based rules are fine-tuned in order to adapt them to a new application domain. A CE system would receive large TRD as an input and would filter it according to CE rules, calculating semantic values. TRD Viewpoint browser is intended to allow a user to browse between the source TRD and partial viewpoint-based views, and to allow further filtering according to application domain criteria. The Semantic filtering broker supports exploitation tools by handling semantic value and rules requests.

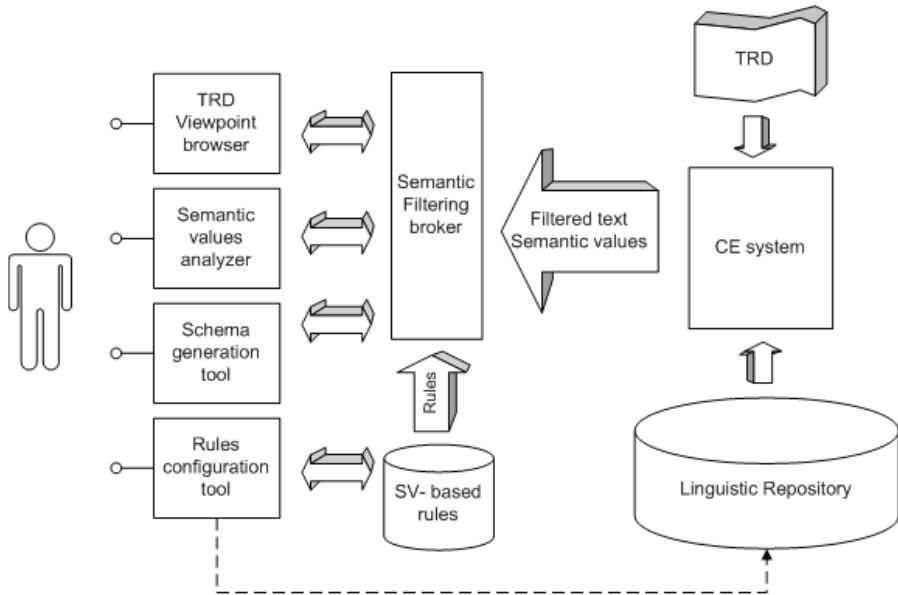


Fig. 1. Architecture for TRD semantic filtering

4 Conclusion and Further Work

This paper has proposed a semantic-based approach for NLRE, which extracts semantically relevant sentences from large TRD making use of linguistic-based rules. It has exposed the CE method, which organises rules, linguistic markers and clues, assigning semantic values according to four major viewpoints, considered as relevant from a RE perspective: relations between concepts, aspecto-temporal organisation, control and causality. Furthermore, this paper has outlined how semantic viewpoints could improve requirements analysis and validation in light of other NLRE approaches.

Currently, work is being done to evaluate the precision of CE rules, markers and clues (most of them issued from linguistic studies on scientific corpus) on industrial requirements documents, as well as on the implementation of a declarative language for semantic-value based rules in a way that could allow inter-operability between viewpoints. Based on other evaluations experiences [15,2] we can conclude that two kinds of evaluation will be necessary in order to know to which extent the proposed approach improve large TRD processing: a linguistic one, which will evaluate the quality of the filtered text fragments, and an empirical one, where real system analyst system analyst could use semantic filtering tools on real TRD.

References

1. V. Ambriola and V Gervasi. Processing natural language requirements. In *12th International Conference on Automated Software Engineering*, pages 36–45, Lake Tahoe, Etats Unis, 1997. IEEE Press.
2. C. Ben-Achour. *Extraction des besoins par analyse de scénarios textuels*. PhD thesis, Université de Paris 6, 1999.
3. J.F.M. Burg. *Linguistic Instruments in Requirements Engineering*. IOS Press, Amsterdam, 1997.
4. Jean-Pierre Desclés. *Langages applicatifs, langues naturelles et cognition*. Hermès, Paris, 1990.
5. Jean-Pierre Desclés. Systèmes d'exploration contextuelle. In C. Guimier, editor, *Co-texte et calcul de sens*, pages 215–232. Presses Universitaires de Caen, France, 1997.
6. F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami. An automatic quality evaluation for natural language requirements. In *Seventh International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'2001)*, 2001.
7. G. Fliedl, C. Kop, and H.C. Mayr. From scenarios to kcpm dynamic schemas: Aspects of automatic mapping. In *Proceedings of the 8th International Conference on Applications of Natural Language and Information Systems (NLDB'2003)*, pages 91–105, Germany, 2003. GI Edition.
8. J. Goguen. Requirements engineering as the reconciliation of social and technical issues. In J. Goguen and M. Jirotka, editors, *Requirements Engineering: Social and technical issues*. Academic Press, London, 1994.
9. A. Jackiewicz. *L'expression de la causalité dans les textes. Contribution au filtrage sémantique par une méthode informatique d'exploration contextuelle*. PhD thesis, Université Paris-Sorbonne, 1998.
10. M. Jackson. *Software Requirements and Specifications : A Lexicon of Practices, Principles and Prejudices*. ACM Press, New York, 1995.
11. B. Kovitz. *Practical Software Requirements*. Manning, London, 1998.
12. P. Laublet, L. Nait-Baha, A. Jackiewicz, and B. Djouia. *Collecte d'information textuelles sur le Web selon différents points de vue*, chapter Interaction homme-machine et recherche d'information. Hermès, Paris, paganelly, c. edition, 2002.
13. Florence LePriot. *Extraction et capitalisation automatiques de connaissances à partir de documents textuels*. PhD thesis, Université Paris-Sorbonne, 2000.
14. L. Mich, M. Franch, and P. Novi Inverardi. Requirements analysis using linguistic tools: Results od an on-line survey. Technical report 66, Department of Computer Management Sciences. Università de Trento, <http://eprints.biblio.unitn.it/view/department/informaticas.html>, 2003.
15. Jean-Luc Minel. *Filtrage Sémantique*. Hermès, Paris, 2002.
16. B. Nuseibeh and S. Easterbrook. Requirements engineering: A roadmap. In A. Finkelstein, editor, *The Future of Software Engineering*. ICSE, Londres, 2000.
17. M. Osborne and C.K. MacNish. Processing natural language software requirement specifications. In *2nd IEEE International Conference on Requirements Engineering*. IEEE Press, 1996.
18. C. Rolland and C. Proix. A natural language approach for requirements engineering. In *Advanced Information System Engineering (Lecture Notes in Computer Science)*. Springer Verlag, Paris, 1992.
19. P. Zave and M. Jackson. Four dark corners in requirements engineering. *ACM Computing Surveys*, 6(1):1–30, 1997.

Author Index

- Abu Shawar, Bayan 407
Alexandrov, Mikhail 229
Atwell, Eric 407
Atzeni, P. 413
- Bagga, Amit 114
Basili, R. 413
Bekhouche, Dalila 380
Besbes Essanaa, Sélima 362
Blanco, Xavier 229
Bolshakov, Igor A. 312, 395
Bontcheva, Kalina 324
Brasethvik, Terje 336
Burton-Jones, Andrew 51
- Calvo, Hiram 207
Campoy-Gomez, Laura 354
Chen, Libo 242
Christodoulakis, Stavros 1
Cimiano, Philipp 401
Cortizo, José Carlos 195
Couchot, Alain 276
Courant, Michele 265
Cunningham, H. 254
- Denis, Xavier 380
Desai, Bipin C. 103
Dix, A. 147
- Fiedler, Gunar 13, 348
Fliedl, Günther 421
Fliedner, Gerhard 64
Frederiks, P.J.M. 123
- Galicia-Haro, Sofía N. 395
García Flores, Jorge J. 427
Gelbukh, Alexander 207, 312, 395
Gómez, José María 195
Gonçalves, Teresa 374
Grilheres, Bruno 380
Grootjen, F.A. 171
Gulla, Jon Atle 217, 336
Guzman-Arenas, Adolfo 182
- Han, Sang-Yong 388
Hansen, D.H. 413
- Hirsbrunner, Beat 265
Horacek, Helmut 26
Huang, Joshua 299
- Kaada, Harald 336
Kang, In-Su 76
Karanastasi, Anastasia 1
Kazasis, Fotis G. 1
Kim, Jintae 159
Kiyavitskaya, Nadzeya 135
Kop, Christian 421
- Lammari, Nadira 362
Lee, Jong-Hyeok 76
Lopez, Vanessa 89
Lydon, S.J. 254
- Ma, Fanyuan 299
Makagonov, Pavel 229
Martí, M^a Antònia 288
Martínez-Barco, Patricio 39
Maynard, D. 254
Mayr, Heinrich C. 421
Mich, Luisa 135
Missier, P. 413
Montoyo, Andrés 288
Motta, Enrico 89
Muñoz, Rafael 39
Mylopoulos, John 135
- Na, Seung-Hoon 76
Nica, Iulia 288
- Olivares-Ceja, Jesus M. 182
Onditi, V.O. 147
- Paggio, Patrizia 413
Park, Sooyong 159
Pazienza, Maria Teresa 413
Pollet, Yann 380
Puertas, Enrique 195
Quaresma, Paulo 374
- Ramduny, D. 147
Ransom, B. 147

- Rayson, P. 147
Rong, Hongqiang 299
Ruiz, Miguel 195
- Salbrechter, Alexander 421
Saquete, Estela 39
Schwanzara-Benoit, Thomas 13
Shin, Kwangcheol 388
Sommerville, Ian 147
Storey, Veda C. 51
Stratica, Niculae 103
Su, Xiaomeng 217
Sugumaran, Vijayan 51, 159
- Tablan, V. 254
Tafat, Amine 265
Thalheim, Bernhard 348
Thiel, Ulrich 242
- Thirunarayan, Krishnaprasad 368
Vázquez, Sonia 288
- Wang, Ziyang 114
Weber, Georg 421
Weide, T.P. van der 123, 171
Wilks, Yorick 324
Winkler, Christian 421
Wolska, Magdalena 26
Wood, M.M. 254
- Yang, Gijoo 76
Ye, Yunming 299
- Zanzotto, Fabio Massimo 413
Zeni, Nicola 135