# INTEGRATION OF WORLD KNOWLEDGE FOR NATURAL LANGUAGE UNDERSTANDING

EKATERINA OVCHINNIKOVA

ATLANTIS THINKING MACHINES
SERIES EDITOR | K.-U. KÜHNBERGER

ATLANTIS PRESS

**ATLANTIS THINKING MACHINES**

VOLUME 3

SERIES EDITOR: **KAI-UWE KÜHNBERGER**

# Atlantis Thinking Machines

Series Editor:

Kai-Uwe Kühnberger

Institute of Cognitive Science
University of Osnabrück, Germany

(ISSN: 1877-3273)

**Aims and scope of the series**

This series publishes books resulting from theoretical research on and reproductions of general Artificial Intelligence (AI). The book series focuses on the establishment of new theories and paradigms in AI. At the same time, the series aims at exploring multiple scientific angles and methodologies, including results from research in cognitive science, neuroscience, theoretical and experimental AI, biology and from innovative interdisciplinary methodologies.

All books in this series are co-published with Springer.

For more information on this series and our other book series, please visit our website at:

*www.atlantis-press.com/publications/books*

ATLANTIS PRESS

AMSTERDAM – PARIS – BEIJING

# Integration of World Knowledge for Natural Language Understanding


**Ekaterina Ovchinnikova**

USC ISI
4676 Admiralty Way
Marina del Rey, CA 90292
USA



ATLANTIS PRESS

AMSTERDAM – PARIS – BEIJING

**Atlantis Press**

8, square des Bouleaux
75019 Paris, France

For information on all Atlantis Press publications, visit our website at: *www.atlantis-press.com*



**Atlantis Thinking Machines**

Volume 1: Enaction, Embodiment, Evolutionary Robotics. Simulation Models for a Post-Cognitivist Science of Mind - Marieke Rohde, Ezequiel A. Di Paolo
Volume 2: Real-World Reasoning: Toward Scalable, Uncertain Spatiotemporal, Contextual and Causal Inference - Ben Goertzel, Nil Geisweiller, Lúcio Coelho, Predrag Janičić, Cassio Pennachin

ISBNs
Print: 978-94-91216-52-7
E-Book: 978-94-91216-53-4
ISSN: 1877-3273

# Foreword

Inference-based natural language understanding (NLU) was a thriving area of research in the 1970s and 1980s. It resulted in good theoretical work and in interesting small-scale systems. But in the early 1990s it foundered on three difficulties:

- Parsers were not accurate enough to produce predicate-argument relations reliably, so that inference had no place to start.
- Inference processes were not efficient enough nor accurate enough.
- There was no large knowledge base designed for NLU applications.

The first of these difficulties has been overcome by progress in statistical parsing. The second problem is one that many people, including Ekaterina Ovchinnikova, are working on now. The research described in this volume addresses the third difficulty, and indeed shows considerable promise in overcoming it. For this reason, I believe Dr. Ovchinnikova's work has a real potential to reignite interest in inference-based NLU in the computational linguistics community.

A key notion in her work is that there already exists sufficient world knowledge in a variety of resources, at a level of precision that enables their translation into formal logic. To my mind, the most important of these are WordNet and FrameNet, especially the latter, and she describes the kind of information one can get out of these resources. She exploits in particular the hierarchical information and the glosses in WordNet, generating 600,000 axioms. She also describes how one can utilize FrameNet to generate about 50,000 axioms representing relations between words and frames, and about 5000 axioms representing frame-frame relations. Her analysis of FrameNet is quite thorough, and I found this part of her work inspiring.

She also critically discusses foundational ontologies such as DOLCE, SUMO. and OpenCyc, and domain-specific ontologies of the sort being constructed for the Semantic Web.

She examines the problems raised by semi-formal ontologies, like YAGO and ConceptNet, which have been gleaned from text or Netizens and which may be more difficult to translate into formal logic. She also shows how to use distributional data for a default mode of processing when the required knowledge is not available.

Her use of knowledge from a variety of resources, combined into a single system, leads to the very hard problem of ensuring consistency in such a knowledge base. She engages in a very close study of the kinds of conceptual inconsistencies that occur in FrameNet and in Description Logic ontologies. She then provides algorithms for finding and resolving inconsistencies in these resources. I found this part of her work especially impressive.

She examines three forms of inference – standard deduction, weighted abduction, and reasoning in description logics, explicating the strengths and weaknesses of each.

Finally she evaluates her work on inference-based NLU by applying her reasoning engines to the Recognizing Textual Entailment problem. She uses the RTE-2 test set and shows that her approach, with no special tuning to the RTE task, achieves state-of-the-art performance. She also evaluates her approach, with similarly excellent results, on the Semantic Role Labeling task and on paraphrasing noun-noun dependencies, both of which fall out as a by-product of weighted abduction.

So the research described here is very exciting indeed. It is a solid achievement on its own and it promises to open doors to much greater progress in automatic natural language understanding in the very near future.

*Jerry R. Hobbs*
*Information Sciences Institute*
*University of Southern California*
*Marina del Rey, California*

# Acknowledgments

The research presented in this book is based on my PhD thesis, that would not have been possible without the help of many people. In the first place I would like to thank my thesis advisor Kai-Uwe Kühnberger who has scientifically and organizationally supported all my research adventures giving me freedom to try whatever I thought was interesting.

I am indebted to Jerry Hobbs who has invited me to visit the Information Sciences Institute where I have spent the most productive six months of my dissertation work. Jerry has introduced me to the exciting field of abductive reasoning and encouraged me to combine this approach with my research efforts, which turned to be highly successful.

I owe my deepest gratitude to Frank Richter who has supported me from the very beginning of my research career. Whenever I needed scientific advice or organizational support, Frank was always there to help.

My gratitude especially goes to the ISI colleagues. I particularly benefited from discussions with Eduard Hovy. Thanks to Rutu Mulkar-Mehta who has developed and supported the *Mini-TACITUS* system, I managed to implement the extensions to the system that many of my research results are based upon. I very much thank Niloofar Montazeri who shared with me the tedious work on recognizing textual entailment challenge.

I thank Nicola Guarino for giving me an opportunity to spend a couple of weeks at the Laboratory of Applied Ontology. Many thanks to the LOA colleagues Laure Vieu, Alessandro Oltramari, and Stefano Borgo for a fruitful collaboration on the topic of conceptual consistency.

The following gratitudes go to the researchers from all around the world who have directly contributed to this work. I very much thank Tonio Wandmacher for being my guide into the world of distributional semantics and for constantly challenging my trust in inference-based approaches. I am grateful to Johan Bos, the developer of the *Boxer* and *Nutcracker* systems, who helped me to organize experiments involving these systems. I would like to thank

Anselmo Peñas for collaborating with me on the issue of paraphrasing noun dependencies. I thank Michael McCord for making the *ESG* semantic parser available for my experiments. I thank Helmar Gust who agreed to write a review of my thesis.

Concerning the financial side, I would like to thank the German Academic Exchange service (DAAD) for according me a three year graduate scholarship. I also thank the Doctorate Programme at the University of Osnabrück for supporting my conference and scientific trips financially.

I would like to thank Johannes Dellert, Ilya Oparin, Ulf Krumnack, Konstantin Todorov, and Sascha Alexeyenko for valuable comments, hints, and discussions. Special thanks to Ilya for keeping asking me when my thesis was going to be finished.

I am grateful to Irina V. Azarova who gave me a feeling of what computational linguistics really is.

I express my particular gratitude to my parents Andrey and Elena for their continued support and encouragement, which I was always able to count on.

Finally, I sincerely thank my husband Fedor who has greatly contributed to the realization of this book. Thank you for valuable discussions, introduction into statistical data processing, manifold technical and software support, cluster programming necessary for large-scale experiments, and all other things, which cannot be expressed by words.

*E. O., November 2011, Los Angeles*

# Contents

**Foreword** v

**Acknowledgments** vii

**List of Figures** xiii

**List of Tables** xv

**List of Algorithms** xvii

**1. Preliminaries** 1

1.1 Introduction . . . 1
1.2 Objectives . . . 4
1.3 How to Read This Book . . . 13

**2. Natural Language Understanding and World Knowledge** 15

2.1 What is Natural Language Understanding? . . . 15
2.2 Representation of Meaning . . . 18
2.2.1 Meaning Representation in Linguistic Theories . . . 19
2.2.2 Linguistic Meaning in Artificial Intelligence . . . 26
2.3 Shared Word Knowledge for Natural Language Understanding . . . 30
2.3.1 Linguistic vs. World Knowledge . . . 31
2.3.2 Natural Language Phenomena Requiring Word Knowledge to be Resolved 33
2.4 Concluding Remarks . . . 36

**3. Sources of World Knowledge** 39

3.1 Lexical-semantic Resources . . . 40
3.1.1 Hand-crafted Electronic Dictionaries . . . 46
3.1.2 Automatically Generated Lexical-semantic Databases . . . 53
3.2 Ontologies . . . 56
3.2.1 Foundational Ontologies . . . 59
3.2.2 Domain-specific Ontologies . . . 64
3.3 Mixed Resources . . . 65
3.3.1 Ontologies Learned from Text . . . 65
3.3.2 Ontologies Learned from Structured Sources: YAGO . . . 66
3.3.3 Ontologies Generated Using Community Efforts: ConceptNet . . . 67
3.4 Concluding Remarks . . . 68

**4. Reasoning for Natural Language Understanding** **73**

4.1 Semantic Parsers . . . 74
4.1.1 English Slot Grammar . . . 74
4.1.2 Boxer . . . 76
4.2 Deduction for Natural Language Understanding . . . 77
4.3 Abduction for Natural Language Understanding . . . 81
4.4 Reasoning with Description Logics . . . 86
4.5 Concluding Remarks . . . 90

**5. Knowledge Base Construction** **93**

5.1 Preliminaries . . . 95
5.1.1 Notation . . . 95
5.1.2 Axiom Weights . . . 95
5.2 Axioms derived from Lexical-Semantic Resources . . . 96
5.2.1 Axioms derived from WordNet . . . 96
5.2.2 Axioms derived from FrameNet . . . 101
5.2.3 Axioms derived from Proposition Store . . . 114
5.3 Ontology . . . 116
5.4 Similarity Space . . . 118
5.5 Concluding Remarks . . . 121

**6. Ensuring Consistency** **123**

6.1 Conceptual Inconsistency of Frame Relations . . . 124
6.1.1 Ontological Status of Frames . . . 126
6.1.2 Constraints on Frame Relations . . . 128
6.1.3 Reasoning-Related Conceptual Problems in FrameNet . . . 131
6.1.4 Case Study . . . 133
6.2 Logical Inconsistency in Ontologies . . . 134
6.2.1 Resolving Logical Inconsistencies . . . 136
6.2.2 Tracing Clashes . . . 139
6.2.3 Resolving Overgeneralization . . . 141
6.2.4 Root and Derived Concepts . . . 145
6.2.5 Rewriting Algorithm . . . 147
6.2.6 Prototypical Implementation . . . 151
6.3 Concluding Remarks . . . 153

**7. Abductive Reasoning with the Integrative Knowledge Base** **155**

7.1 Adapting *Mini-TACITUS* to a Large Knowledge Base . . . 155
7.2 Refining Abductive Reasoning Procedure for NLU . . . 162
7.3 Reasoning with Ontologies . . . 166
7.4 Reasoning with Similarity Space . . . 171
7.5 Concluding Remarks . . . 173

**8. Evaluation** **177**

8.1 Natural Language Understanding Tasks . . . 179
8.1.1 Recognizing Textual Entailment . . . 179
8.1.2 Semantic Role Labeling . . . 182
8.1.3 Paraphrasing of Noun Dependencies . . . 185
8.2 Experiments with *Boxer* and *Nutcracker* . . . 187

8.2.1 Recognizing Textual Entailment . . . 189
8.2.2 Semantic Role Labeling . . . 194
8.3 Experiments with *Mini-TACITUS* . . . 195
8.3.1 Recognizing Textual Entailment . . . 196
8.3.2 Semantic Role Labeling . . . 201
8.3.3 Paraphrasing of Noun Dependencies . . . 202
8.3.4 Domain Text Interpretation . . . 207
8.4 Concluding Remarks . . . 212

**9. Conclusion** **215**

**Appendix A** **221**

**Bibliography** **225**

**Index** **241**

# List of Figures


1.1 Inference-based NLU pipeline. . . . . . . . . . . . . . . . . . . . . . . . . . . 9

2.1 Human natural language understanding. . . . . . . . . . . . . . . . . . . . . . . 17
2.2 Computational natural language understanding. . . . . . . . . . . . . . . . . . . 18
2.3 Meaning of *John gave Mary a book* as a conceptual dependency graph. . . . . 28

3.1 Illustration of the definition for the term “ontology” given by Guarino (1998). . 57

4.1 Inference-based NLP pipeline . . . . . . . . . . . . . . . . . . . . . . . . . . 73
4.2 The LF produced by *ESG* for the sentence *If Mary gives John a book, then he reads it.* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
4.3 The LF produced by *ESG* for the sentence *The book is given to John by Mary.* . 75
4.4 The DRS produced by *Boxer* for the sentence *If Mary gives John a book, then he reads it.* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76

5.1 Modules of the proposed integrative knowledge base. . . . . . . . . . . . . . . . 94
5.2 Schematic overview on the generation of an LSA-based semantic space. . . . . 120

6.1 DOLCE basic categories. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 127
6.2 “Medical” cluster: frame relations from FrameNet (top) enriched and cleaned up (bottom). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 133

7.1 Abductive reasoning pipeline. . . . . . . . . . . . . . . . . . . . . . . . . . . 175

8.1 NLU pipeline based on *Boxer* and *Nutcracker*. . . . . . . . . . . . . . . . . . 190
8.2 NLU pipeline based on *ESG* and *Mini-TACITUS*. . . . . . . . . . . . . . . . . 196

# List of Tables

3.1 English WordNet 1.3 statistics . . . . . . . . . . . . . . . . . . . . . . . . . . 48
3.2 English FrameNet 1.5 statistics. . . . . . . . . . . . . . . . . . . . . . . . . . 52
3.3 VerbOcean statistics. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55

4.1 Syntax and semantics of $\mathcal{ALCN}$ DL. . . . . . . . . . . . . . . . . . . . . . . 87
4.2 Tableau expansion rules for $\mathcal{ALCN}$ satisfiability. . . . . . . . . . . . . . . . 88

5.1 Direction of entailment for WordNet relations. . . . . . . . . . . . . . . . . . . . 99
5.2 Axioms derived from WordNet morphosemantic relations. . . . . . . . . . . . 100
5.3 Statistics for axioms extracted from WordNet. . . . . . . . . . . . . . . . . . . . 101
5.4 Results of FrameNet frame clustering. . . . . . . . . . . . . . . . . . . . . . . 113
5.5 Statistics for axioms extracted from FrameNet. . . . . . . . . . . . . . . . . . . 113

6.1 Tableau expansion rules for tracing clashes in $\mathcal{ALCN}$ terminology. . . . . . 140

8.1 Examples of text-hypothesis pairs from the RTE-2 development set. . . . . . . 181
8.2 Evaluation of the *Shalmaneser* system towards FATE. . . . . . . . . . . . . . . 184
8.3 Results of recognizing textual entailment by the *Nutcracker* system for the 39 RTE-2 pairs annotated with “medical” frames in FATE. . . . . . . . . . . . . . 194
8.4 Evaluation of the *Boxer* system performing SRL towards FATE. . . . . . . . . 194
8.5 Evaluation of the *Mini-TACITUS* system performing RTE using RTE-2 test dataset. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 201
8.6 Evaluation of the *Mini-TACITUS* system performing SRL towards FATE. . . . 202
8.7 Paraphrasing of noun dependencies in the RTE-2 development set. . . . . . . . 205
8.8 Paraphrasing of noun dependencies in the RTE-2 test set. . . . . . . . . . . . . 205
8.9 Paraphrasing axioms used to resolve RTE-2 development set pairs. . . . . . . . 205
8.10 Paraphrasing axioms used to resolve RTE-2 test set pairs. . . . . . . . . . . . . 208

# List of Algorithms


6.1 Repair terminology $\mathscr{T}$ containing unsatisfiable concept $C$. . . . . . . . . . . . . 148
7.1 *Mini-TACITUS* reasoning algorithm: interaction of the time and depth parameters. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 158
7.2 Algorithm for selecting axioms, which can be evoked given a logical form. . . . 159

# Chapter 1

# Preliminaries

## 1.1 Introduction

In order to understand a natural language expression it is usually not enough to know the literal ("dictionary") meaning of the words used in this expression and compositional rules of the corresponding language. Much more knowledge is actually involved in discourse processing; knowledge, which may have nothing to do with the linguistic competence but is rather related to our general conception of the world. Suppose we are reading the following text fragment.

> *"Romeo and Juliet" is one of Shakespeare's early tragedies. The play has been highly praised by critics for its language and dramatic effect.*

This piece of text is perfectly understandable for us, because we can relate its meaning to our general knowledge about culture and everyday life. Since we know that the most famous Shakespeare was a playwright and the main occupation of playwrights is writing plays, we conclude that the word *tragedy* in this context refers to a work of art rather than to a dramatic event and that Shakespeare has written it rather than, for example, possessed. The time attribute *early* can refer only to an event, therefore we infer that it modifies the event of Shakespeare's writing "Romeo and Juliet". Time attributes of art creation events are usually defined relative to the life time of the corresponding creators. Therefore we conclude that Shakespeare has written "Romeo and Juliet" when he was young. Knowing that a tragedy is a kind of play, we can relate *"Romeo and Juliet"* to *the play* in the next sentence. Similarly, the knowledge about plays being written in some language and having dramatic effect helps to resolve the anaphoric *its*.

Over all its history, semantics, and especially computational semantics, was puzzled with the question of how to construct a formal and possibly unambiguous representation of the linguistic meaning. In contemporary linguistic theory, one of the dominating views on semantics has been put forwards by the framework called formal semantics, which has been largely inspired by Montague (1973). In this framework, a sentence meaning is given in terms of a logical representation, also sometimes called *logical form*.[1] Logical representations allow us to abstract from some of the syntactic features of a sentence and make its logical features explicit in order to enable proper inferences, which otherwise would not follow. For example, a unique logical representation will be assigned to the active/passive alternations. The sentences *Shakespeare wrote a tragedy* and *A tragedy was written by Shakespeare* will get the same representation: $\exists t,s,e(tragedy(t) \wedge Shakespeare(s) \wedge write(e,s,t))$.[2]

Constructing logical representations, as abstractions from the surface form of utterances, is an essential step on the way to grasping the linguistic meaning. However, for semantics this is not the end of the story, because there is a whole bunch of pragmatic problems in discourse processing, which cannot be solved on the basis of linguistic knowledge isolated from knowledge about the world. These are such problems as reference resolution, interpretation of noun compounds and prepositional phrases, resolution of some kinds of syntactic and semantic ambiguity, metonymy resolution, just to name a few. Concerning the example above, the logical representation does not help us to decide whether *tragedy* in this context refers to a dramatic event or to a work of art. This is where lexical semantics comes into play, which covers theories of the classification and decomposition of word meaning, and the relationship of word meaning to sentence meaning and syntax, cf. Cruse (1986). In a lexical semantic framework, the word *tragedy* will be most probably defined as ambiguous, having at least two meanings, such that its first meaning is related to the words like *catastrophe*, and its second meaning is related to *play*.

Naturally, different approaches to enrich logical representation with world knowledge have arisen. Some of these approaches are based on *reasoning*; in practice they employ an inference machine.[3] In this framework, world knowledge is represented in the form of axioms, which constitute a *knowledge base* (KB). Given the logical representation and the KB, the inference machine can answer queries and solve particular tasks such as detecting inconsistencies in the logical representation, distinguishing between new and already

[1] See Sec. 2.2.1 for more details on the term *logical form*.

[2] If not stated differently, all formulas are given in first order logic notation. Logical forms are represented using a Davidsonian approach (Davidson, 1967), so that the first argument of every verb predicate is an event variable, the second argument is a prototypical agent, the third argument is a prototypical theme/patient, and the forth argument is a prototypical goal/recipient, see Dowty (1991) for a description of the prototypical roles.

[3] Alternative approaches are mentioned in Chap. 2.

known information, deciding whether one text piece logically entails another one, etc. For example, given two sentences *"Romeo and Juliet" is one of Shakespeare's early tragedy* and *Shakespeare is the author of "Romeo and Juliet"*, the inference machine can be used to infer that the first sentence logically entails the second one. Concerning natural language processing (NLP), this type of reasoning is intended to facilitate such applications as, for example, question answering, information extraction, and dialog systems.

In spite of the strong theoretical basis and practical relevance of the inference-based approaches to semantics, only a few of them result in successful NLP systems and even fewer – in systems applicable to "real-life" texts going beyond manually constructed examples. There are particular reasons for it. First, inference-based approaches are currently given little attention, which is mostly diverted to statistical approaches dominating in natural language processing. Application of logical inferencing to NLP requires a lot of interdisciplinary knowledge from the domains of formal semantics, lexical semantics, logics, knowledge representation, and automated reasoning. As Bos (2009) points out, "only few researchers feel comfortable in the intersection of these areas, and collaborations between researchers of these fields are rare". Second, inference-based approaches are still computationally expensive, which prevents their efficient application to large amounts of data. Third, there are still open research questions and problems in these approaches, which make showing a good performance on "real-life" text data difficult.

One of such sticking points concerns building and using the knowledge base. Although the practical success of an inference-based NLP system crucially depends on the quantity and quality of the data stored in the KB, the current research on automated reasoning as applied to NLP focuses much more on treating logically complex constructions (e.g., quantification) than on equipping the developed inference machine with world knowledge. The issue of constructing a knowledge base might seem to be purely practical. However, it appears to pose several fundamental research questions. Some of them can be formulated as follows.

What sources of world knowledge are appropriate for construction of a knowledge base suited for natural language understanding, e.g., human conceptualization of the world, collections of documents, dictionaries? How to formalize collected knowledge and to convert it into axioms readable by inference machines? If knowledge was taken from different sources, then how to create a reasonable mapping of the vocabulary? How to ensure consistency of the KB? After knowledge has been collected and integrated into a consistent KB, how to organize an interface between the logical representations and axioms in the

KB in order to enable application of axioms? How to make an inference machine work efficiently with the large KB?

The questions listed in the paragraph above outline the central topic of this book. In particular, the book is concerned with three main issues. First, it discusses development of a consistent knowledge base designed for natural language understanding, which has a sufficient lexical and conceptual coverage and will enable inferences required for NLP on a large scale. Second, it discusses reasoning procedures able to make use of this knowledge. Finally, it presents several experiments intended for the evaluation of the KB and the reasoning procedures in computational natural language understanding. The main goal of the book is to show that inference-based approaches to semantics have good potential to be successfully applied to processing "real-life" text data on a large scale and are able to compete with purely statistical methods dominating in the area of NLP so far.

## 1.2 Objectives

In the following, we will go into more details of the three directions of work described in this book: construction of a knowledge base, development of a reasoning pipeline, and evaluation of the proposed approach.

### *Constructing a Knowledge Base*

At the present time, there are two main types of knowledge bases used in the NLP community. First, there are lexical-semantic resources, which contain information about classification of word senses and semantic relations defined on word senses, such as, for example, the "type-of" relation, e.g., *dog* is a type of *animal*. Traditionally, lexical-semantic knowledge was represented by hand-crafted electronic dictionaries, like, for example, WordNet (Fellbaum, 1998b) which has become doubtlessly the most popular lexical-semantic resource in the NLP community. Recently, with the development of the statistical approaches to NLP and growing size of the available corpora, more and more researchers became interested in automatic extraction of the lexical-semantic information.

The idea of applying statistical methods for generation of lexical-semantic knowledge comes back to the *distributional hypothesis* claiming that words occurring in the same contexts tend to have similar meanings (Harris, 1954). In this framework, distributional features of words (contexts, in which they occur in a corpus) are used to compute similarity distances between them, for example, *dog* might appear to be more similar to *cat* than to *table*. These similarity distances can then be used to measure semantic overlap between two text pieces, which facilitates solving semantic problems such as reference resolution, recognizing textual entailment, disambiguation, and many others. For examples of inference-based approaches relying on lexical-semantic knowledge, see (Tatu and Moldovan, 2005; Bos and Markert, 2006).

Another type of knowledge used in NLP is provided by *ontologies*. In artificial intelligence, an ontology is a formal representation of knowledge of a given domain. It is represented by a set of concepts and a set of relationships between these concepts (Gruber, 2009). Ontological knowledge is usually given by axioms formalized in an expressive logic specially designed for reasoning (see Sec. 3.2 for more details). One example of an ontological axiom is $\forall x(\textsc{Pacific\_island}(x) \rightarrow \textsc{Island}(x) \wedge \exists y(\textsc{locatedIn}(x,y) \wedge \textsc{Pacific\_ocean}(y)))$, which says that a Pacific island is an island located in the Pacific ocean. The main purpose of such ontologies is to enable reasoning over their content, i.e. to enable the reasoner to answer queries about concepts, relations, and their instances. In practice, ontologies usually contain detailed and structured knowledge about particular domains (e.g., geography, medicine).[4] Ontologies are less popular in NLP than lexical-semantic resources. This is probably due to the fact that ontological vocabularies consist of concept labels instead of words (e.g., FRENCH_POLYNESIA_ISLAND instead of *French Polynesia island*) and mapping these labels to natural language lexicons and logical representations creates an additional non-trivial problem. Nevertheless, there are NLP systems, which have successfully overcome this difficulty and managed to employ ontological knowledge, see, for example, (Franconi, 2002; Frank *et al.*, 2005).

There are some overlaps and redundancies between existing lexical-semantic dictionaries, automatically generated lexical-semantic databases, and ontologies. But for the most part, the mentioned knowledge sources contain disjoint information. Hand-crafted lexical-semantic dictionaries provide both an extensive lexical coverage and a high-value semantic labeling. However, such resources lack certain features essential for capturing some of the knowledge required for linguistic inferences. First of all, manually created resources are in principle static; updating them with new information is a slow and time-consuming process. By contrast, common sense knowledge and the lexicon are dynamic entities, which are subject to continuous changes. New words appear and existing words acquire new senses and associations.[5] For accommodating dynamic knowledge, static knowledge bases

[4]However, there exist general purpose ontologies as well (see Sec. 3).

[5]This especially concerns proper names, which constantly emerge and disappear from our shared knowledge background.

seem to be not the best solution. On the contrary, automatically learned resources seem to be ideal for representing frequently changing data. They can be easily updated using any kind of corpora, for example, the World Wide Web, which reflects the most recent state of the target language. However, automatic learning unavoidably implies noisy data with a poor semantic labeling as compared to the databases built manually. Therefore, at least at the moment, automatically generated knowledge bases cannot fully substitute hand-crafted resources, but can successfully extend them, as we will show in Chap. 7.

Lexical-semantic resources as a knowledge source for reasoning have a significant shortcoming: They imply too little structure. In most of the cases, a semantic relation is just a two-place predicate ($relation\_name(word_1,word_2)$), and there is not much space in this representation for defining complex relationships like, for example, the fact that a Pacific island is an island located in the Pacific ocean. Although semantic relations constitute a solid basis for reasoning, they appear to be far from sufficient for enabling all inferences required for text interpretation. A more elaborated axiomatization of word senses and their combinations is needed in many cases.[6] This axiomatization can be accommodated by ontologies supporting an expressive representation language and sophisticated inference techniques.

What we propose in this book is a reasoning framework, which accommodates all these types of the existing knowledge bases (hand-crafted lexical-semantic resources enriched with automatically generated semantic relations and ontologies). Different types of knowledge are stored in separate modules of an integrative knowledge base, which supports inferences with the KB inside of one single reasoning pipeline. The pipeline queries modules separately and unifies the obtained information on every reasoning step. The developed knowledge base is intended to have sufficient lexical and conceptual coverage in order to be applicable to both general types of texts (e.g., newspaper articles) and domain-specific documents (e.g., medical histories).

There are two reasons of why we are advocating the modular organization of the KB. The first reason concerns substitutability of modules. Lexical-semantic knowledge obviously depends on the target language; existing lexical-semantic databases are for the most part domain-independent. In contrast, ontological knowledge is mostly language-independent, but it may be strongly focused on a particular domain. Keeping lexical and ontological knowledge apart will enable easier module replacement as well as switching to another language or application domain.

[6]Examples can be found in (Clark *et al.*, 2007).

The second reason is related to inferences implied by different types of knowledge. At this point, we will distinguish between commonsense and "scientific" knowledge. The commonsense knowledge (which is closely related to our lexical competence, because we unavoidably capture everyday concepts with words) is for the most part fuzzy, probabilistic, and non-definitional. Indeed, instead of strict definitions people seem to conceive lexical meaning in the form of associations. Inconsistencies, circular "definitions" and imprecisions are perfectly accepted. Plenty of psycholinguistic studies support this claim (cf. Murphy, 2002). In general, handling ambiguities and fuzziness seems to be unproblematic for human beings in different areas of their everyday life. Therefore, an inference procedure intended to handle commonsense knowledge must tolerate inconsistencies and borderline cases as well as it must rely on probabilistic information. In contrast, when people try to understand how the world is organized in order to build a theory, they seek to arrive at non-ambiguous and precise definitions passing tests on consistency. Ontologies usually imply a kind of a theory in background (even if the theory is implicit) and are explicitly intended to be logically consistent and unambiguous (for more details see Sec. 3.2). Therefore, a specific type of reasoner should be employed for handling ontologies. The modular organization of the knowledge base enables us to use different reasoning mechanisms for the different types of knowledge as we will show in Chap. 7.

An additional aspect in constructing a knowledge base concerns ensuring consistency of the KB, which is crucial for drawing correct inferences. The two types of consistency considered in this book are logical and conceptual consistency. Logical consistency concerns logical contradictions in the axioms, which result in the unsatisfiable KB (see Sec. 6). Logical contradictions can arise only if the axioms are expressive enough to contain negation. Relations derived from lexical-semantic resources involve no negation, therefore the problem of logical consistency affects ontologies only. A lot of literature in the field of reasoning with ontologies concerns detecting and repairing inconsistencies, see, for example, Kalyanpur (2006). In this book, we present a new automatic procedure for repairing logically inconsistent ontologies. The procedure eliminates logical contradictions and is maximally knowledge-preserving.

Conceptual consistency is related to formal ontological foundations of modeling the world. When classifying the world into concepts and giving definitions for these concepts, knowledge engineers cannot avoid being subjective, because there exist yet no generally accepted standards in the field of knowledge representation. Without a background theory of ontological modeling, it is easy to make conceptual mistakes, which will result in misleading inferences. In the area of overlap of computer science and philosophy, a few researchers have been concerned with formulating ontological principles for designing taxonomies, which have had success in the community of ontology engineers, see (Guarino and Welty, 2004). There were attempts to apply these principles to lexical-semantic resources such as WordNet, see (Oltramari *et al.*, 2002). In this book, we consider some other types of conceptual relations going beyond taxonomy such as causation, temporal precedence, etc., and we worked on formulating ontological principles for modeling these relations. Instead of focusing on atomic concepts (e.g., the concept of *book*), as it has been done in the previous studies, we consider more structured entities, which are *frames* – predicates with the fixed number of defined arguments (e.g., *book*(*author*, *language*, *topic*)).[7] This study is carried out with a particular focus on NLP applications, i.e. the ontological principles formulated for modeling conceptual relations were aimed at enabling inferences relevant for natural language understanding.

To sum up, the proposed knowledge base integrates lexical-semantic resources and ontologies in a consistent and modular way. It enables us to accommodate different types of knowledge in one reasoning framework, relatively easy switching between languages and domains of knowledge, and providing extensive lexical and conceptual coverage.

### *Reasoning with Integrative Knowledge Base*

As already mentioned above, inference-based approaches to natural language semantics rely on a semantic parser outputting logical representations and on a knowledge base providing axioms for reasoning. A generalized inference-based NLP pipeline is schematized in Fig. 1.1. A text fragment is first fed to a parser, which outputs its logical representation. The logical representation and the knowledge base constitute input for the inference machine. An external NLP application generates queries for the inference machine (e.g., it asks it to check whether a logical formula follows from the KB) and uses its output to solve specific tasks. The three processing components of this pipeline (parser, inference machine, and final application) are usually independent from each other and can be replaced by alternative systems. In the study described in this book, we are not concerned with developing a new semantic parser, but use some of the existing systems (for more details see Chap. 4.1). Similarly, we are not aiming at developing a specific NLP application. The proposed reasoning procedure as well as the obtained knowledge base are evaluated in three tasks, namely recognizing textual entailment, semantic role labeling, and paraphras-

[7]For more details on frames see Sec. 2.2.2 and 3.1.

ing noun dependencies (see Chap. 8), but the presented approach remains fairly general and is not tuned for any particular application. This book rather focuses on the question of how to design a reasoning procedure able to handle a KB containing heterogeneous knowledge (as described above) on a large scale.

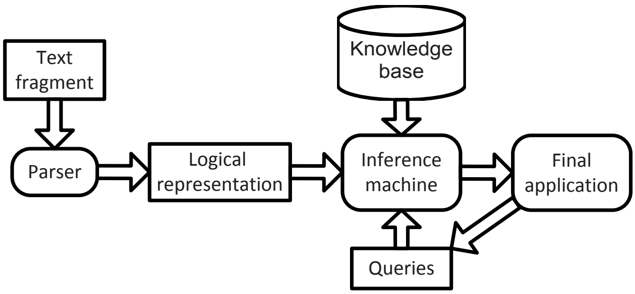


Fig. 1.1 Inference-based NLU pipeline.

The idea of applying automated reasoning to natural language understanding originated in the context of automated question answering, see, for example, Winograd (1972) for an overview of the early approaches. At present, there exist two main development directions for the inference-based approaches to semantics. The first one, which has a longer history and still gets more attention in computational semantics, is based on deduction (see Blackburn *et al.*, 2001, for an overview). In the second line of research, abduction is employed as the principal reasoning mechanism (see, for example, Hobbs *et al.*, 1993).

*Automated deduction* implies automatic proving of formulas given a theory. Deductive inference machines try to prove whether a formula is valid, i.e. whether it is entailed by a given logical system. In natural language understanding, the task is to check whether a logical representation is consistent with the knowledge base or/and follows from the knowledge base and possibly to build a model satisfying the logical representation. In practice, deductive reasoners are mainly used for a) detecting logical inconsistencies in text representation, b) distinguishing between new and already known information, c) building models satisfying logical representations, d) given two representations, detecting whether one of them logically entails the other one. For example, if the knowledge base contains the axiom $\forall x(dog(x) \rightarrow animal(x))$ and there is a text fragment *Pluto is a dog* with the logical representation $\exists p(Pluto(p) \wedge dog(p))$ then the deductive reasoner should be able to infer that $\exists a(animal(a))$ follows from this text fragment.

*Abduction* is inference to the best explanation. If $a$ logically implies $b$ then given an observation $b$, an abductive reasoner will assume $a$ as a possible explanation of $b$. When abduction is used for discourse processing, the process of interpreting text fragments is viewed as the process of providing the best explanation of why these text fragments would be true given the knowledge base. Thus, the reasoner tries to "prove" parts of the input logical representation by anchoring them to the information given in the knowledge base and to each other. For example, given the sentence above about Pluto followed by the sentence *This animal is funny* with the logical representation $\exists a(animal(a) \wedge funny(a))$ an abductive reasoner should link *dog* in the first sentence with *animal* in the second sentence, so that the variables $p$ and $a$ are set to be equal, because, given the information about an existence of an animal, an existence of a dog may be assumed. There might be many possible "proofs" in this procedure. A special kind of abduction called *weighted abduction* allows us to define a cost function on proofs, which enables us to chose the "best" (the cheapest) proof.

In the experiments described in this book, we used both a deductive and an abductive reasoner. The natural language understanding system based on deduction, which we used is called *Nutcracker* (Bos and Markert, 2006). Unfortunately, the experiments with *Nutcracker* showed that using automated deduction is not the optimal solution for accomplishing our goals. In most of the cases the system has failed to find a proof, because some (maybe very little) piece of the relevant knowledge was missing. On the contrary, the reasoning procedure based on weighted abduction is robust with respect to missing knowledge and has some other advantages (see Sec. 4.5 for more details). Therefore, we incorporated the obtained integrative knowledge base into an abductive inference procedure (see Sec. 7 for more details). The abductive reasoner, which we extend in order to make it able to employ distributional information and ontological knowledge is called *Mini-TACITUS*, see Hobbs *et al.* (1993) for the theoretical background.

As already mentioned above, reasoning with ontologies requires a special type of inference machines, which can handle the formalism underlying the ontology representation language. Probably the most popular one of existing markup languages for ontology design is the Web Ontology Language (OWL; McGuinness and van Harmelen, 2004), which is based on the logical formalism called Description Logic (DL; Baader *et al.*, 2003). The recent progress of the Semantic Web technologies[8] has stimulated extensive development of the OWL ontologies as well as development of DL-based reasoners.[9] In the presented

[8]*http://www.w3.org/2001/sw/*
[9]See, for example, *http://www.cs.man.ac.uk/∼sattler/reasoners.html*.

experiments, we integrated one of the existing DL-reasoners into the abductive inference machine (see Sec. 7).

Thus, the proposed reasoning pipeline is based on a weighted abduction algorithm extended to make use of probabilistic lexical-semantic knowledge and equipped with a DL-reasoner for querying ontologies.

***Experimental Evaluation***

We evaluated the developed inference procedure as well as the obtained knowledge base in several natural language understanding tasks the main of which is recognizing textual entailment (RTE). As the reader will see in the following chapters, the proposed reasoning pipeline is fairly general and not tuned for a particular application. We decided to test the approach on RTE because this is a well-defined task that captures major semantic inference needs across many natural language processing applications, such as question answering, information retrieval, information extraction, and document summarization.

In the RTE task, the system given a text and a hypothesis must decide whether the hypothesis is entailed by the text plus common sense knowledge. For example, given the text sentence *"Romeo and Juliet" is one of Shakespeare's early tragedies* and the two hypotheses *Shakespeare is the author of "Romeo and Juliet"* and *Shakespeare went through a tragedy* the system should predict that the first hypothesis is entailed by the text and the second one is not.

In 2004, Dagan, Glickman, and Magnini have started a series of competitions under the PASCAL Network of Excellence, known as the PASCAL Recognizing Textual Entailment Challenges (Dagan *et al.*, 2005). Every RTE Challenge provided researches with a development set for training and a test set for evaluating the participating systems. All RTE datasets are freely available to the community. For evaluation, we used one of these data sets.

The first RTE experiment described in this book was done with the deductive reasoning system. This particular experiment shows some of the typical problems in automatic reasoning as applied to natural language and demonstrates that axiomatized lexical-semantic resources can serve as a good basis for building a knowledge base for reasoning (see Sec. 8.2). In the further experiments, we employed the abductive reasoning pipeline.

In addition to RTE, we evaluated how well the procedure does in assigning semantic roles. Semantic role labeling (SRL) is one of the most popular tasks in the field of computational semantics. It consists in detection of the semantic arguments associated with the predicate and their classification into specific roles. For example, in the sentence *John gave Mary a book* the verb *give* has three arguments: *John*, *Mary*, and *book*. The task is to annotate the arguments with appropriate semantic role labels. The argument *John* might be annotated with *Donor*, *Agent*, or $Arg_1$ depending on the underlying linguistic theory.

As the reader will see in Chap. 5, a part of the integrative knowledge base consists of the axioms extracted from the FrameNet lexical-semantic resource, which enables us to annotate text with FrameNet semantic roles. Therefore, for the proposed approach, evaluation of SRL appeared to be relatively straightforward given an appropriate gold standard. As a gold standard we used the Frame Annotated Corpus for Textual Entailment (Burchardt and Pennacchiotti, 2008), which consists of one of the RTE test sets annotated with semantic roles.

Further more, we performed an experiment on paraphrasing noun dependencies such as noun compounds and possessives. The problem of paraphrasing such dependencies consists in finding an appropriate relation explicitly expressing the dependency. For example, *morning coffee* is most probably interpreted as a coffee **drunk** in the morning, while *morning newspaper* most probably stands for a paper **read** in the morning. In the experiment on paraphrasing noun dependencies, we considered those noun-noun constructions, which are relevant for computing entailment in one of the RTE datasets.

Performing the described evaluation tests (RTE, SRL, and paraphrasing of noun dependencies), we aimed at showing that 1) it is in principle possible to design a reasoning procedure capable to handle a heterogeneous large knowledge base, and 2) the knowledge base helps to improve system performance. We did not intend to outperform existing systems specializing in these tasks. Most of the systems participating in RTE challenges are equipped with RTE-specific heuristics helping them to solve this particular task, while the presented system was not. Similarly, systems performing SRL are usually trained on special semantic role annotated corpora. This was not the case for the presented system. Nevertheless, in the RTE and SRL tasks the proposed abduction-based reasoning procedure showed performance compatible with those of the state-of-the art systems. Since the reasoning procedure and the integrative knowledge base are general and were not adapted for these particular tasks, we consider the results of the experiments to be promising concerning possible manifold applications of the developed pipeline.

In addition, we tested our reasoning pipeline as applied to interpretation of a domain-specific text. Since there exist no standard test sets for evaluating domain-specific NLU, we built a small test set from scratch and performed a qualitative rather than a quantitative study of the proposed procedure. In this experiment, we employed a small ontology from the domain of cars and semi-automatically collected entailment pairs belonging to the same domain from the web. The performed experiment revealed strong and weak sides of the proposed approach and outlined future work directions.

## 1.3 How to Read This Book

The first chapters of this book are aimed at giving a sufficient background in the areas relevant for NLU. Furthermore, in these chapters, we covered problematic and controversial issues and provided a critical analysis of the existing frameworks, methods, and resources related to NLU. We hope that this material might be useful to researchers interested in an introduction into computational NLU. Chapter 2 gives an introduction to natural language understanding (Sec. 2.1) and overviews existing approaches to representation of linguistic meaning (Sec. 2.2). Furthermore, it discusses the differences between lexical and world knowledge, and provides an overview of linguistic phenomena requiring world knowledge for their resolution (Sec. 2.3).

In Chap. 3, a brief description of the most influential of the existing machine readable resources containing world knowledge is given. Section 3.1 is devoted to lexical-semantic resources. In Sec. 3.2, the two types of ontologies are considered: foundational and domain-specific ontologies. Section 3.3 is devoted to mixed resources, which cannot be definitely classified, because they have features of both ontologies and lexical-semantic resources. The chapter is concluded with a comparative analysis of the presented knowledge resources with respect to natural language understanding (Sec. 3.4).

Chapter 4 gives an overview of tools and methodologies for the application of automatic reasoning to natural language understanding. Section 4.1 describes two semantic parsers employed in the experiments described in this book. Section 4.2 introduces automated deduction as applied to natural language understanding. Section 4.3 concerns weighted abduction in natural language understanding with a particular focus on the approach presented in Hobbs *et al.* (1993). This approach builds the theoretical basis for the system called *Mini-TACITUS*, which we extended for enabling it to make use of the integrative knowledge base. Section 4.4 introduces reasoning with Description Logics, which is a special case of deductive reasoning. The chapter is concluded with a comparative analysis of deduction and weighted abduction with respect to natural language understanding (Sec. 4.5).

Chapter 5 describes, which resources were included into the integrative knowledge base and how axioms for reasoning were generated from these resources. Chapter 6 concerns ensuring consistency of the developed knowledge base. Conceptual consistency is in focus of Sec. 6.1, where relations defined on structured concepts (frames) are considered. In Sec. 6.2, a new algorithm for resolving logical inconsistency in ontologies is presented. Chapter 7 describes extensions of the abductive reasoning system *Mini-TACITUS* intended to make it work using the developed integrative knowledge base. Optimization steps allowing the system to treat large amounts of data are described in Sec. 7.1 and 7.2. Sections 7.3 and 7.4 focus on the extensions of *Mini-TACITUS* enabling integration of reasoning with ontologies and similarity spaces into the abductive inference procedure.

In Chap. 8, the evaluation results are presented. Section 8.1 introduces natural language understanding tasks, on which the evaluation was based. The experiments with the semantic parser *Boxer* and the deductive reasoning system *Nutcracker* are described in Sec. 8.2. Section 8.3 presents evaluation of the *Mini-TACITUS* system equipped with the developed integrative knowledge base.

Chapter 9 concludes the book and gives an outlook on the further research perspectives of the inference-based NLU.

## Chapter 2

# Natural Language Understanding and World Knowledge

### 2.1 What is Natural Language Understanding?

In artificial intelligence and computational linguistics, *natural language understanding* (NLU) is a subfield of natural language processing that deals with machine reading comprehension. The goal of an NLU system is to *interpret* an input text fragment. The process of interpretation can be viewed as a translation of the text from a natural language to a representation in an unambiguous formal language. This representation, supposed to express the text's content, is further used for performing concrete tasks implied by a user request.

One of the earliest known attempts in natural language understanding is the program *STUDENT* developed by Bobrow (1964) for his dissertation project. This was a program, which could solve high-school algebra problems posed in a natural language form. For example, *STUDENT* was able to answer the following query: "If the number of customers Tom gets is twice the square of 20% of the number of advertisements he runs, and the number of advertisements is 45, then what is the numbers of customers Tom gets?" *STUDENT* was processing queries by doing simple pattern-matching in order to translate the queries into a format suitable for an equation-solving program.

A couple of years later, Weizenbaum (1966) created *ELIZA*, an interactive computer program able to carry on a dialog on any topic. *ELIZA* was equipped with a simple parser; its "semantic" analysis was based on transformation of key words into canned patterns. Although *ELIZA* was just a toy project, it has gained an amazing popularity, especially in the area of psychotherapy. The popularization of *ELIZA* and similar programs has led to the appearance of the term "ELIZA effect"[1] describing the tendency of people to attribute human-like intelligence to chat robots. Indeed, a typical dialog with *ELIZA* may look quite impressive:

[1] See preface in Hofstadter (1996) for a description of ELIZA effect.

```
 USER: Men are all alike
ELIZA: In what way?
 USER: They're always bugging us about something or other
ELIZA: Can you think of a specific example?
 USER: My boyfriend made me come here
ELIZA: Your boyfriend made you come here
 USER: He says I'm depressed much of the time
ELIZA: I'm sorry to hear you are depressed
...
```

*STUDENT* and *ELIZ*A were among the first computer programs to approach human intelligence in language comprehension and they were followed by many others. But can we actually say that these programs were able to *understand* natural language? How can we judge it?

The "Holy Grail" for most AI applications is a human-like performance. In order to get a clearer picture of what we expect from the computer systems, let us consider a human language understanding scenario. The man on Fig. 2.1[2] asks the woman to give him a pen for writing a letter. She analyses his request applying available knowledge and tries to guess what kind of meaning he has in mind. Finally, she concludes that he is asking for a writing instrument (rather than, for example, for a portable enclosure, in which babies may be left to play) and passes it over to him. Can we say that the woman in this scenario understands the message? She correctly complies the speaker's request, but it could be done by chance. What would probably convince us in her understanding is a comparison of the interpretation of the message, which she has created with the meaning intended by the speaker. Unfortunately, for humans such a comparison is impossible, because human conceptualizations are not explicitly available. There are only indirect ways to evaluate understanding, namely, by comparing the interpreter's behavior with the predicted behavior.

A similar scenario is applicable to computer programs. If a program is doing what we expect it to do then we can say that it "understands" the input. For example, if a search system outputs information that we have been searching, we can say that the system "understands" our query. If a summarization system summarizes a document like a human would do it then we can say that the system "understands" the document.

[2]The pictures of the pens on Fig. 2.1 are provided by DesignContest (*http://www.designcontest.com*) under Creative Common licence vers. 3 (*http://creativecommons.org/licenses/by/3.0/*).

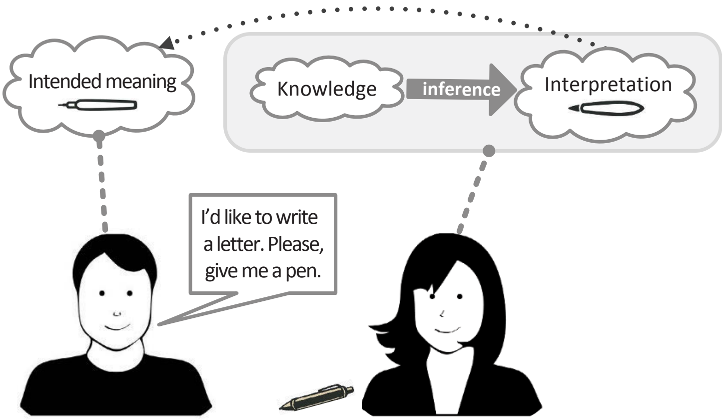


Fig. 2.1 Human natural language understanding.

For computer programs, there also exists another criterion of understanding. Working with computer programs, in contrast to human agents, we can look inside of processing and see what representation of the message content was created by the program. This automatically created representation can then be evaluated against human intuition about the content of the processed text fragment. Probably neither two speakers will arrive to the same conceptualization of the message content, because of the differences in individual experience. Thus, there will be as many conceptualizations as the number of readers comprehending the text fragment. However, all these conceptualizations will probably have a common part implied by the shared linguistic and conceptual knowledge of a language community rather than by individual experience. This shared part of the conceptualization is what we want our NLU system to grasp. The more information occurring in the shared conceptualization it can represent, the better it "understands" the text fragment, see Fig. 2.2 for an illustration.

Thus, there are two criteria for judging how well an NLU system "understands" a text fragment. The first criterion is performance-based. This evaluation strategy is realized in such series of test challenges as, for example, Text Analysis Conference (TAC[3]). The organizers of this challenge provide a large test collections for different NLP tasks and common evaluation procedures, which enable comparison of the output of NLP systems against hu-

[3] *http://www.nist.gov/tac/*

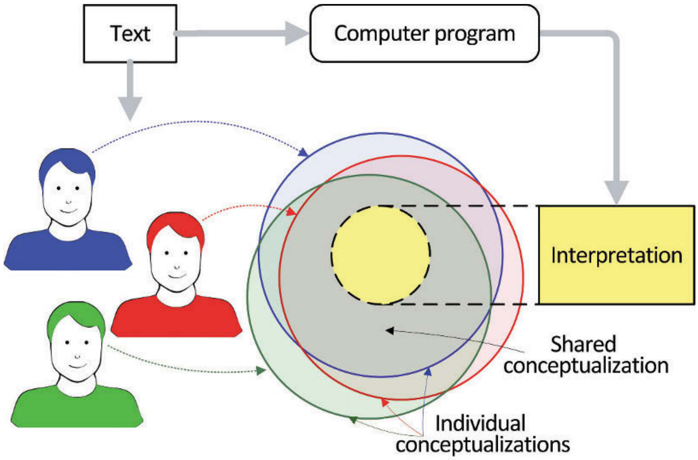


Fig. 2.2 Computational natural language understanding.

man solutions for the same tasks. The traditional TAC evaluation tasks are knowledge base population, recognizing textual entailment, and summarization.

The second evaluation criterion takes into account the intern representation of the text content created by the system. For testing NLP programs according to this criterion, automatically generated representations of a text are compared with human annotations of the same text. This evaluation strategy is realized, for example, in the serious of competitions called Semantic Evaluation (SemEval[4]) which mostly focus on such NLP tasks as word sense disambiguation, semantic role labeling, coreference resolution, subcategorization acquisition, etc.

## 2.2 Representation of Meaning

Talking about representations of the text content, one should regard the question about what an adequate representation of linguistic meaning should look like, which information it should carry, and how it can be constructed. There is no unique answer to these ques-

[4] *http://aclweb.org/aclwiki/index.php?title=SemEval_Portal*

tions; it depends on the underlying theory of linguistic meaning. Only abstract claims can be made about meaning representations in general. For example, the assumptions concerning representation of linguistic meaning made by Schank (1972) and Allen (1987) can be summarized as follows.

(1) If two sentences have the same/closely related meaning, they should be represented the same.
(2) The representation must be precise and unambiguous enough to enable us to express every distinct reading of a sentence as a distinct formula.
(3) Information implicitly stated in the sentence should be represented explicitly.

In this section, we review prominent approaches to natural language semantics in the fields of linguistics and artificial intelligence. Our goal is to discuss **what representations** were considered as suitable for expressing linguistic meaning and knowledge required for language understanding and **what information** was considered to be a part of meaning of natural language expressions. Special attention is payed to the **role of world knowledge** in different theories of linguistic meaning.

Before starting the discussion about theories of linguistic meaning, one needs to make a remark on terminology. Many of the basic terms in natural language semantics are highly ambiguous and presuppose a diverse understanding in different frameworks. Probably the most ambiguous term is “meaning” itself. One of the most influential definitions of this term was suggested by Frege who claimed that meaning is a relation between a sign, an object in the world, to which the sign refers, and a corresponding concept in someone’s mind. Nowadays, not all researchers working on natural language semantics accept this definition. Especially in computational linguistics, the term “meaning” is often understood in its most general sense, as a relation between a linguistic sign and some non-linguistic entity, to which the sign refers; it can be a concept in the mind, an object in the world, a logical formula, etc. Since the work presented in this book does not rely on one single theoretical framework, but rather adopts ideas originating in totally different research paradigms, we adopt this fairly general understanding of “meaning”.

### 2.2.1 *Meaning Representation in Linguistic Theories*

#### 2.2.1.1 *Formal Semantics*

In the 1970’s, natural language semantics came under the influence of Montague’s work (Montague, 1973) proposing *formal semantics*, which represents the first formal “implementable" approach to linguistic meaning. In this framework, the focus has been set on the logical properties of natural language. Formal semantics mainly concentrates on the syntax-semantics interface, defining rules, which allow us to translate surface structures into logical representations in a compositional way. This approach is also called *model-theoretic semantics*, because it accounts for linguistic meaning in terms of truth conditions (hypothetical states of affairs, which the sentence correctly describes) and denotation (objects in the world words refer to).

In formal semantics, a sentence meaning is given by its logical representation, sometimes called *logical form*[5], which is a formal representation of its logical structure derived from the corresponding surface form. For example, the sentences *Shakespeare wrote a tragedy* and *A tragedy was written by Shakespeare* will be assigned the same logical representation, which abstracts from the surface form of the sentences: $\exists t,s,e(tragedy(t) \wedge Shakespeare(s) \wedge write(e,s,t))$.[6]

This approach mainly concentrates on linguistic means of expressing logical features of a natural language expression, such as, for example, quantification, logical connectors, or modality. The meaning of the non-logical predicates (e.g., *Shakespeare*, *write*, *tragedy*) expressed by content words, as opposed to function words (e.g., *and*, *if*, *a*), is irrelevant in the context of formal semantics. It is defined in a referential way, i.e. the meaning of *tragedy* is given by the set of all entities, which can be referred to as "tragedy". Thus, the sentences *a cat eats a rat* and *a mat facilitates a nap*, which have the same syntactic structure, will be assigned the logical representations equal to $\exists x,y,e(P(x) \wedge Q(y) \wedge R(e,x,y))$.[7] Distinguishing between these sentences is then a matter of an interpretation function mapping $P$, $Q$ and $R$ to different sets.

Discourse semantics has extended the Montague's approach in order to go beyond sentence boundaries. In this framework, it is possible to represent the semantic structure of a sequence of sentences describing different eventualities. For example, Discourse Representation Theory (Kamp and Reyle, 1993) and Dynamic Predicate Logic (Groenendijk and Stokhof, 1991) consider intersentential anaphora and relations between the eventualities of

[5]The term *logical form* is rather controversial. In traditional linguistics, it is strongly associated with the framework of generative grammar (see, for example, Chomsky, 1976; May, 1977; Heim, 1982), which is not related to formal semantics and not concerned with inferences following from natural language sentences. In computational semantics and artificial intelligence, the term is understood in a wide sense; it refers to *any* formal representation of the text content given as a logical formula (see, for example, Hobbs *et al.*, 1993; Bos, 2008; McCord, 2010). In the following chapters of this book, we use the term *logical form* in the latter sense.

[6]This representation corresponds to a Davidsonian approach (Davidson, 1967). For simplicity, tense is disregarded in this and the following examples.

[7]This example is taken from Vieu (2009).

different sentences. This approach has been developed in Segmented Discourse Representation Theory (SDRT) (Asher, 1993; Lascarides and Asher, 1993; Asher and Lascarides, 2003). In order to link discourse segments, SDRT uses temporal, causal, mereological, or argumentative *discourse relations*.[8] The SDRT research naturally focuses on explicit discourse connectors, e.g., *but*, *because*. In addition, SDRT considers how world knowledge affects discourse structure. For example, knowledge like "when something is pushed, normally, it moves" enables us to establish the temporal and causal relations between such discourse segments as *John fell. Max pushed him*.[9]

Formal semantics has gained widespread popularity both among linguists and computer scientists, because it has opened new ways of computing natural language meaning. But it is definitely not the end of the story for computational semantics, because many natural language phenomena require more knowledge for their resolution than just logical structure.[10] For example, the logical representation does not help us to decide whether the word *tragedy* in the example above refers to a dramatic event or to a work of art. Being able to resolve this ambiguity presupposes looking deeper into the intern meaning of the content words. This was done by *lexical semantics* in its different versions.

#### 2.2.1.2 *Lexical Semantics*

Lexical semantics developing in the framework of generative grammar and structuralism considered lexical meaning to be a starting point for a semantic theory (Katz and Fodor, 1963; Jackendoff, 1972). The main paradigms involved decomposing lexical meaning into *semantic markers* – atomic units of meaning and conceptualization. For example, Katz and Fodor (1963) proposed to capture different senses of the noun *bachelor* in terms of such semantic primitives as *human/animal*, *male*, *young*, *who has never been married*, *who has the first or lowest academic degree*, etc. This theory distinguishes between dictionary (definitional) and encyclopedic knowledge. While the former is considered to be a part of the lexical meaning, the latter is not. For example, the attribute *being a vehicle* is a part of the meaning of *car*, while *moving on a road* is not, because the latter attribute is prototypical rather than necessary. In practice, it turned out to be difficult to find "definitions" and a finite set of semantic markers for the largest part of the lexicon. Shortly after it appeared, the Katz-Fodor theory was subjected to diverse criticisms (see, for example, Bolinger, 1965; Vermazen, 1967; Putnam, 1975).

[8] The notion of discourse relations is borrowed from Rhetorical Structure Theory (Mann and Thompson, 1988).
[9] This example is taken from Asher and Lascarides (2003).
[10] See Sec. 2.3.2 for detailed examples.

Probably, the most successful application of the decomposition approach concerns verb meanings as decomposed into thematic roles. In this approach, the lexical meaning of a verb includes a specification of the types of arguments associated with this verb. For example, the verb *put* has an associated “putter” (agent role), the thing that is put (theme role), and the place where it is put (location role). This approach was taken, for example, by Fillmore (1968), Jackendoff (1987), and Dowty (1991). Nowadays, decomposition of verb meanings into thematic roles seems to be a standard solution for verb semantics. However, this approach still has a fundamental problem concerning fixing a universal inventory of roles and the ambiguity in assigning roles (see Riemer, 2010, for an overview).

Another aspect of verb meaning elaborated in the framework of generative grammar concerns *selectional preferences*, which are the semantic constraints that a word imposes on the syntactic environment. For example, the verb *to drink* usually takes a beverage as its object. This knowledge can help to disambiguate sentences like *Mary drank burgundy*, while *burgundy* can be interpreted as either a color or a beverage.[11] Chomsky (1965) has incorporated selectional preferences into his syntactic theory, whereas other researchers considered them to be predictable from lexical meanings (e.g., McCawley, 1973).

Instead of a definition-based model of lexical meaning, Rosch (1978) has proposed Prototype theory considering a category as consisting of different elements, which have unequal status. For example, a robin is a more prototypical member of the class of birds than a penguin and *being able to fly* is a more prototypical attribute of birds than *eating worms*. Prototype theory is based on Rosch’s psycholinguistic research on internal structure of categories (Rosch, 1975). The conclusion followed from the experiments involving response times, priming, and exemplar naming was that some members/attributes of a category are more privileged than others. Prototype theory has quickly attracted a lot of attention, but also a lot of criticism. The critical issues concern problems of identifying attributes for classes, accounting for category boundaries, treating abstract non-visual categories, and compositionality issues (see Riemer, 2010, for an overview).

In spite of the problematic issues, prototype models of categorization have had a significant influence on the research in semantics. Many of the insights of prototype research are accounted for in cognitive approaches to semantics, which aim at developing a comprehensive theory of mental representation. The term “cognitive semantics” covers a variety of different approaches, which share several common points. Cognitive semantics proposes a holistic view on language. Cognitivists like Langacker (1987) and Lakoff (1987)

[11] This example is provided by Resnik (1997).

reject the modular approach promoted by Chomsky (1965) assuming that language is one of a number of independent modules or faculties within cognition. Consequently, most researchers in cognitive semantics reject the dictionary–encyclopedia distinction (see, for example, Jackendoff, 1983; Langacker, 1987). As a result, semantic knowledge like "bachelors are unmarried males" is considered to be not distinct from encyclopedic knowledge. In cognitive semantics, lexical meaning has a conceptual nature; it does not necessarily concern a reference to an entity in the world. Instead, meaning corresponds to a concept held in the mind, which is related to other concepts such that without knowledge of all of them, one does not have complete knowledge of any one.

Different cognitive approaches to semantics propose different models of the structure of concepts underlying lexical meaning. Fillmore's ideas have developed into Frame semantics (Fillmore, 1968), which considers lexical meanings to be related to prototypical situations captured by *frames* – structures of related concepts. Lakoff (1987) has introduced *idealized cognitive models*, which are theories of particular domains reflected in language. Langacker (1987) has developed Cognitive Grammar modeling semantic aspects as *image schemes*. Talmy has published a number of influential works on linguistic imaging systems (e.g., Talmy, 1983, 2000). Being quite popular among researchers working in the overlap area of linguistics and psycholinguistics, cognitive semantics has been mainly criticized for informality and the speculative character of cognitivist theories not really grounded on psychological experiments (see Riemer, 2010, for an overview of the critics).

As an alternative to classical decomposition theory of meaning and cognitive semantics, a relational approach to lexical meaning has been developed. Instead of defining lexical meaning in terms of semantic primitives, the meaning is represented as a network of relationships between word senses called *lexical-semantic relations*.[12] For example, one sense of bachelor is related to *unmarried* and another sense is related to *academic degree*. This approach has been described in detail by Cruse (1986). In computational linguistics, it has been implemented in electronic network-like dictionaries, the most famous of which is currently WordNet (Miller *et al.*, 1990; Miller and Fellbaum, 1991; Fellbaum, 1998b). Similar to decomposition theories, semantic networks describe lexical meaning in a definitional way. For example, WordNet relates *airplane* to *vehicle*, but not to *sky*. In contrast to decomposition theories, in this approach words are defined by other words rather than by semantic primitives. In a relational framework, representation of such complex definitions as "who has the first or lowest academic degree" is impossible, because

[12] See Sec. 3.1 for more details on lexical-semantic relations.

each lexical-semantic relation is just a two-place predicate relating word senses. However, this representation simplicity makes this approach implementable and extremely useful in practical NLP.

A relatively recent theory of linguistic meaning called Generative Lexicon (GL) was proposed by Pustejovsky (1991, 1995). Pustejovsky criticized the standard view of the lexicon, on which each lexeme is associated with a fixed number of word senses. For example, the adjective *fast* implies three different senses in the phrases *a fast typist* (one who types quickly), *a fast car* (one, which can move quickly), and *a fast waltz* (one with a fast tempo).[13] Pustejovsky argues that just listing these senses does not help to account for creative language use. For example, the use of *fast* in *fast motorway* cannot be accounted on the basis of the senses mentioned above. In order to cope with this problem, Pustejovsky focuses on additional non-definitional aspects of lexical meaning. He introduces semantic structures, which he calls *qualia structures* of words. A qualia structure includes facts about the constituent parts of an entity (Constitutive role), its place in a larger domain (Formal role), its purpose and function (Telic role), and the factors involved in its origin (Agentive role). For example, the qualia structure of *school* includes *an educational institution* as its Telic aspect and *building* as its Formal aspect. This knowledge enables us to generate different senses of *school* in such sentences like *The school was painted white* and *John has learned it at school*. The Generative Lexicon theory represents an important step towards linking lexical meaning to world knowledge. However, the theory has weak points concerning the speculative character of the qualia roles and the difficulty of assigning these roles to concepts associated with a target lexeme, for experimental studies revealing weak points in GL; see Kilgarriff (2001); Cimiano and Wenderoth (2007) .

Formal and lexical semantics refer to quite orthogonal aspects of linguistic meaning. Formal semantics accounts for logical features of languages, pays particular attention to compositionality, and focuses mainly on functional words, while content words are represented as atomic predicate names having referential meaning. In contrast, lexical semantics mostly ignores logical aspects, does not propose any adequate theory of compositionality, and concentrates on the specification of the lexical meaning of content words. It seems to be natural that both approaches could perfectly supplement each other in an integrative approach enabling a fuller understanding of natural language meaning. However, up to the

[13]This example is provided by Pustejovsky (1995).

present time not so many researchers have been working in the both frameworks; formal and lexical semantics seem to a large part to ignore each other.[14]

#### 2.2.1.3 *Distributional Semantics*

With the development of machine learning techniques, distributional approaches to lexical meaning have become extremely popular in computational linguistics and practical NLP. These approaches are based on the idea captured in the famous quotation from Firth (1957): "You shall know a word by the company it keeps". This idea is often referred to as the *distributional hypothesis*, because it presupposes deriving lexical meaning from the distributional properties of words: "words which are similar in meaning occur in similar contexts" (Rubenstein and Goodenough, 1965).

The distributional hypothesis is often considered to originate in the works of Harris (1954, 1968). In this approach, linguistic meaning is inherently differential, and not referential; differences of meaning correlate with differences of distribution. Distributional semantics defines lexical meaning of a word $w$ as a vector of values of similarity between $w$ and other words in the corpus. There are different approaches to calculating co-occurrence-based semantic similarity between two words $w_1$ and $w_2$.

One approach is based on pointwise mutual information (PMI) defined as:

$$log_2 \frac{freq(w_1, w_2)}{freq(w_1) \cdot freq(w_2)},$$

where $freq(w_1, w_2)$ is the frequency of co-occurrence of $w_1$ and $w_2$ and $freq(w_i)$ is the frequency of occurrence of $w_i$. Pointwise mutual information was introduced as a lexical association norm by Church and Hanks (1989). The authors showed that word pairs with a high PMI are often semantically or associatively related.

Another approach to semantic similarity is based on the *vector space models*. In these models, a word $w$ is represented by a vector of word co-occurrence frequencies. Each vector dimension $k$ shows how many times $w$ co-occurs with another word $w_k$ in the same context. The context can be defined as a sequence of $n$ words, entire document, a fixed pattern (e.g., *X is a part of Y*), or a syntactic structure. The similarity of two words is captured as the distance of the corresponding vectors, which can be calculated by one of the usual vector distance measures (e.g., Euclidean distance, cosine).[15] The comparison of co-occurrence vectors is also referred to as *second order co-occurrence* (cf. Grefenstette,

[14] But see integrative approaches developed by Pustejovsky (1995), Partee and Borschev (1998), Asher and Lascarides (2003).
[15] An overview on different distance measures is given by Salton and McGill (1986).

1994). In this approach, similarity is established because two words occur with similar words rather than with each other. Thus, two words can prove to be semantically related even if they never co-occur.

Different models for computing vector-based semantic similarity have been developed. The most prominent approaches include Hyperspace Analogue to Language (Lund and Burgess, 1996), Latent Semantic Analysis (Landauer, 2007), Topic-based vector space model (Kuropka and Becker, 2003), Generalized vector space model (Tsatsaronis and Panagiotopoulou, 2009).

Naturally, distributional semantics does not make any distinction between lexical and world knowledge. All possible associative links, which can be mined out of corpora are considered to be a part of lexical meaning. In this approach, the word *airplane* can be related to *vehicle*, *fly*, *pilot*, *plane crash*, *Aerobus*, and *Wright brothers*.

Distributional semantics provides an account of compositionality by assessing the acceptability of verb-noun, adjective-noun, noun-noun combinations. For example, the higher similarity between *boil* and *potato* as compared to the pair *boil* and *idea* can be used to predict that the combination *boil a potato* is more acceptable than the combination *boil an idea*. Based on this idea, semantic similarity has been used for modeling selectional preferences (Resnik, 1997; Erk *et al.*, 2010; Schulte im Walde, 2010) and learning qualia structures as defined in the Generative Lexicon theory (Lapata and Lascarides, 2003b).

In psycholinguistics, the distributional hypothesis has been used to explain various aspects of human language processing, such as lexical priming (Lund *et al.*, 1995), synonym selection (Landauer and Dumais, 1997), and semantic similarity judgments (Mcdonald and Ramscar, 2001). It has also been employed for a wide range of NLP tasks, such as disambiguation, information retrieval, anaphora resolution, identification of translation equivalents, word prediction and many others.[16]

### 2.2.2 *Linguistic Meaning in Artificial Intelligence*

The earliest natural language understanding programs, e.g., *STUDENT* and *ELIZA* described in Sec. 2.1, were able to process specific predefined domains only. Input sentences were restricted to simple declarative forms and were scanned by the programs for predefined key words and patterns.

Some of the systems developed during the mid-1960s (see, for example, Raphael, 1964; Craig *et al.*, 1966; Collins and Quillian, 1969), were able to store text representations and

[16]See Manning and Schtze (1999) for an overview.

draw simple inferences. Given the sentences *Sokrates is a man* and *All men are mortal*, they could answer queries like "Is Sokrates mortal?" These systems were using formal representations for storing information in a database and employed simple semantic processing for translating input sentences into this representation. Some systems, for example, could use simple logical relations like "if *A* is a part of *B* and *B* is a part of *C*, then *A* is a part of *C*", but this relationship had to be stored in the program, so that they could only handle relationships they were designed for.[17]

Most of the natural language understanding programs developed in the 70s and later might be called *knowledge-based* systems; their development is closely related to the AI research on knowledge representation. These programs use world knowledge about the domain of interest, which is required for text interpretation. Knowledge-based systems can be roughly classified according to representation schemes and reasoning mechanisms which they employ to access world knowledge.

#### 2.2.2.1 *Procedural Semantics*

In the framework of procedural semantics, knowledge is represented as an executable program in a computer language. Both meaning of sentences and knowledge about the world are represented in this way. The execution of these programs corresponds to reasoning from the meanings and knowledge. Thus, in procedural semantics, meaning is embodied in abstract procedures for determining referents, verifying facts, computing values, and carrying out actions. These procedures are built on computational operators and can include sensing and acting in the world. In the early 70s, two systems were developed, which were employing procedural semantics. Winograd's *SHRDLU* was verbally controlled by a user and simulated a robot manipulating simple objects (Winograd, 1972). Woods's *LUNAR* system answered queries about the samples of rock brought back from the moon (Woods *et al.*, 1972). For example, *LUNAR* represented the query "What is the average concentration of Aluminum in each breccia?" as a little program:

```
(FOR EVERY X5 / (SEQ TYPECS) : T ;
  (PRINTOUT (AVGCOMP X5
    (QUOTE OVERALL) (QUOTE AL203))))
```

[17] A good overview of the early NLU systems in given by Winograd (1972).

#### 2.2.2.2 *Semantic Networks*

Quillian (1968) proposed a knowledge representation framework named *semantic networks*, which quickly became popular and has been employed by a variety of knowledge-based NLU systems (see Sowa, 1987, for an overview). Semantic networks, being a model for human associative memory, represented word and sentence meanings as a set of nodes linked in a graph. Networks were used to represent both meaning of text fragments and world knowledge. Quillian has developed simple operations on semantic networks that corresponded to drawing inferences. Compared to formal logical deduction, this sort of reasoning appeared to be more simple and efficient.

Inspired by semantic networks and the dependency theory of Hays (1964), Schank (1972) developed *conceptual dependency* theory. Figure 2.3 represents a conceptual dependency graph for the sentence *John gave Mary a book*. Schank used different kinds of arrows for different relations, such as $\Leftrightarrow$ for the agent-verb relation. The distinction between the semantic network theory and the conceptual dependency theory lies in their focus. Semantic networks are about how knowledge should be organized and how to interpret a semantic net structure. This approach says nothing about what should be represented. In contrast, conceptual dependency theory aims at enumerating the types of nodes and arcs used to build meaning representations. This theory specifies content rather than structure. The conceptual dependency representation was used by Schank and his colleagues in several NLU systems (Schank, 1975; Schank and Abelson, 1977).

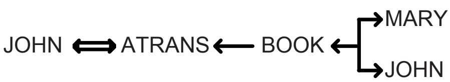


Fig. 2.3 Meaning of *John gave Mary a book* as a conceptual dependency graph.

#### 2.2.2.3 *Frames*

Minsky (1975) proposed a knowledge representation format based on *frames*:

> When one encounters a new situation (or makes a substantial change in one's view of the present problem) one selects from memory a structure called a Frame. This is a remembered framework to be adapted to fit reality by changing details as necessary.
>
> A frame is a data-structure for representing a stereotyped situation, like being in a certain kind of living room, or going to a child's birthday party. Attached to each frame are several kinds of information. Some of this information is about how to use the frame. Some is

about what one can expect to happen next. Some is about what to do if these expectations are not confirmed.

For example, the concept of *Pacific island* can be represented by the following frame: `[is-a: island, located: Pacific_ocean, belongs_to: country, name: island_name]`, where `is-a`, `located`, `belongs_to`, and `name` are predefined slots for characterizing islands. In this framework, linguistic meaning is given by mapping of linguistic constituents into corresponding frame slots. Reasoning over frames is based on a unification procedure, which Minsky (1975) defines as follows: "given two frames *A* and *B*, [...] *A* can be viewed as a kind of *B* given a "mapping" or frame-transformation *C* that expresses (perhaps in terms of other mappings) how *A*'s terminals can be viewed in terms of *B*'s terminals".[18]

Building upon this framework, Schank and Abelson (1977) introduced the concepts of scripts, plans, and themes to handle story-level understanding. The classical example of Schank's theory is the restaurant script, which has the following characteristics:

Scene 1: Entering
S PTRANS S into restaurant, S ATTEND eyes to tables, S MBUILD where to sit, S PTRANS S to table, S MOVE S to sitting position

Scene 2: Ordering
S PTRANS menu to S (menu already on table), S MBUILD choice of food, S MTRANS signal to waiter, waiter PTRANS to table, S MTRANS 'I want food' to waiter, waiter PTRANS to cook

Scene 3: Eating
Cook ATRANS food to waiter, waiter PTRANS food to S, S INGEST food

Scene 4: Exiting
waiter MOVE write check, waiter PTRANS to S, waiter ATRANS check to S, S ATRANS money to waiter, S PTRANS out of restaurant

A variety of computer programs have been developed to implement the theory. Schank (1991) applied his theoretical framework to story telling and the development of intelligent tutors. Schank and Cleary (1995) described an application of these ideas to educational software.

In the late 70s and 80s, frame-based knowledge representation was one of the most active area of AI research in natural language understanding; see Barr (1980) for an overview of the early systems. The Ontological Semantics framework (Nirenburg and Raskin, 2004) is an example of a more recent large long-term project employing frame-like structures for

[18]For an elaborated approach to unification, see Shieber (1986).

knowledge representation. This approach represents an attempt on combining linguistic analyses (syntax, semantics-pragmatics pipeline) with background knowledge.

#### 2.2.2.4 *Logical Formulas*

The idea of representation of linguistic meaning by logical formulas and using automated deduction for natural language understanding originated in the context of automated question answering. Green and Raphael (1968) developed a system that offered the full expressiveness of the first-order predicate calculus for the representation of natural language meaning. The deductive procedure of this system was based on an automatic theorem-proving algorithm by Robinson (1965). In this framework, both linguistic meaning and the knowledge base were represented as a set of logical axioms.

At present, there exist two main development directions for the approaches to NLU, which employ theorem proving. The first one follows the initial ideas of using automated deduction, see Sec. 4.2 for more details. In the second line of research, abduction rather than deduction is employed as the principle reasoning mechanism, see Sec. 4.3.

Some of the recent deduction-based approaches are using the full expressiveness of first order logic (see, for example, Dahlgren *et al.*, 1989; Bos and Markert, 2006). Others employ a decidable fragment of FOL, such as Description Logics, see Franconi (2003) for an overview of the applications of Description Logics to natural language processing.

## 2.3 Shared Word Knowledge for Natural Language Understanding

In the previous section of this chapter, we considered how representation of the linguistic meaning has been approached in linguistics and artificial intelligence. We have seen that many researchers working on computational NLU have come to the conclusion that world knowledge associated with content words plays an important role in understanding natural language. A natural question arises about what the term “world knowledge” actually means in the context of NLU and whether/how world knowledge differs from linguistic knowledge. Section 2.3.1 concerns this issue. Section 2.3.2 gives an overview of the typical natural language phenomena requiring world knowledge for their resolution, which represent the main challenges for knowledge-intensive NLP.

One important remark should be made at this point. Obviously, any kind of non-linguistic knowledge may be useful for the goals of natural language understanding: knowledge about a specific domain, which is considered in text, about laws of nature (e.g., physical laws), about the state of the art in the world, about the specifics of the text producer and his current state and so on. In this book, we focus on knowledge that is **shared** by humans belonging to the same linguistic and cultural community and do not consider situational and individual aspects of discourse.

### 2.3.1 *Linguistic vs. World Knowledge*

In order to discuss possible differences between knowledge of language and knowledge about the world, we illustrate different levels of knowledge relevant for natural language understanding with the following examples.[19]

(2.1) If *NP* is a noun phrase and *V* is an intransitive verb, then the concatenation *NP V* is a clause.
(2.2) The phrase *x wrote y* corresponds to the proposition $write(x,y)$.
(2.3) The proposition $write(x,y)$ refers to a "text creation" event, such that $x$ plays the role of author and $y$ plays a role of text in this event.
(2.4) If something is a play then it is a dramatic composition.
(2.5) The main function of a playwright is writing plays.
(2.6) If $x$ creates $y$ at time $t$ then $y$ is an artifact and it has not existed before $t$.
(2.7) "Romeo and Juliet" was written by Shakespeare.

Example (2.1) represents a typical syntactic rule. In Ex. (2.2), a surface realization of the predicate *write* is mapped to a logical form. In Ex. (2.3), the predicate and its arguments are mapped to an event frame and corresponding roles. Example (2.4) describes the *type-of* relation. Example (2.5) refers to the common sense knowledge about playwrights. Features of artifacts are defined in Ex. (2.6). Example (2.7) contains a specific fact about the world.

Syntactic rules like (2.1) are included in the grammar and are language-dependent. Mappings from surface realizations to logical forms like (2.2) are often a part of the lexicon, also language-dependent. Rules like (2.3) and (2.4) can be included into a lexical-semantic dictionary like FrameNet and WordNet.[20] Knowledge like (2.5) can occur in a definition provided by a thesaurus (e.g., WordNet), in a common sense ontology (e.g., OpenCyc), or in a lexicon (e.g., a lexicon based on the Generative Lexicon theory). Statements like (2.6) can be included into an abstract ontological theory of artifacts or into event calculus

[19]These examples are for a large part inspired by similar examples provided by Hobbs (2009). The reason for us constructing new examples instead of directly citing Hobbs is that we intend to link them to the theories and resources described elsewhere in this book.
[20]All the mentioned resources are discussed in Chap. 3

semantics. Facts like (2.7) can be a part of a factual ontology containing knowledge about instances rather than classes (e.g., YAGO).

It is quite straightforward that rules like (2.1) and (2.2) are language-dependent and belong to linguistic knowledge, while (2.7) is not related to the linguistic competence, it is a part of knowledge about the world. Everything between (2.2) and (2.7) is more difficult to classify. Statements (2.3)-(2.6) concern lexical knowledge, i.e. knowledge about word meanings, which is both language-dependent and anchored to the world.

Different semantic theories consider different types of knowledge to be part of lexical meaning. For example, according to the lexical-semantic theories as presented by Cruse (1986) and Miller and Fellbaum (1991), the lexical meaning of *play* comprises (2.4). Frame semantics (Fillmore, 1968) concerns knowledge like (2.3) to be a part of the lexical meaning of *write*. In the framework of Generative Lexicon (Pustejovsky, 1991), knowledge like (2.5) is an integral part of the lexical meaning of *playwright*.

Drawing a line between lexical semantics and world knowledge is a difficult issue. However, some researchers believe that this distinction is important. As the reader has seen in the previous section, this view has been especially promoted by the researchers working in the framework of traditional structuralism and generative grammar. Recently, it is supported by computational linguists working on formal grammars. For example, Copestake (1992) claims that it would help to isolate linguistic theory from non-linguistic phenomena:

> [I]t is methodologically important to distinguish between linguistic and non-linguistic representation, even though the two have to be interrelated so that linguistic utterances can be interpreted as having some connection with the real world. We want to avoid the situation where linguistic representation is dependent on the scientific knowledge about the world [...] we wish to provide a testable constrained theory, and a formal representation language, and to avoid problems, which arise in knowledge representation which do not have a linguistic dimension.

In Copestake's view, in order to construct a lexicon one should "start from the null hypothesis that all that the lexicon contains [...] are pointers connecting the phonological or orthographic representation of the word with its real world denotation. We then have to establish criteria for providing further information about word meaning, which will ensure that the additions have linguistic motivation" (Copestake, 1992). It remains unclear, however, whether establishing such criteria is possible at all.

Other researchers do not consider the borderline between lexical and world knowledge to be crucial for an adequate theory of linguistic meaning. For example, researchers working in the framework of cognitive semantics reject the dictionary-encyclopedia distinction.

More recently, Hobbs (2009) claims that "lexical knowledge is just ordinary knowledge where the entities in question are words". In order to support this view, the author reviews some of the psycholinguistic studies (e.g., Hagoort *et al.*, 2004; Tanenhaus and Brown-Schmidt, 2008) suggesting that semantic interpretation cannot be separated from non-linguistic knowledge. Hobbs argues that "[t]he most common argument in linguistics and related fields for drawing a strict boundary between lexicon and world is a kind of despair that a scientific study of world knowledge is possible". As opposed to this despair, Hobbs suggests a scientific account of world knowledge and a framework, in which all levels of semantic interpretation can be equally implemented as inferences.

In line with the cognitive approach to semantics, in this book, we do not distinguish between lexical and world knowledge. Thus, we are concerned with world knowledge as exemplified by the statements like (2.3)-(2.7), i.e. everything which goes beyond syntax and mapping of surface predicate-argument constructions to predications. Although the integrative knowledge base proposed in this book stores different types of knowledge in separate modules (cf. Chap. 5), this happens due to various technical reasons and not because we believe that there is a cognitive motivation for such modularity.

### 2.3.2 *Natural Language Phenomena Requiring Word Knowledge to be Resolved*

*Ambiguity*

The potential ability of linguistic signs to have more than one meaning is one of the major problems in NLP. Ambiguity affects all linguistic levels: phonological, morphological, lexical, syntactic, and semantic. Text fragments lifted out of context can be highly ambiguous, whereas within a discourse ambiguity can be mostly successfully resolved with the help of context and background knowledge. Example (2.8) below is a classical example of the syntactic ambiguity: Did John use a telescope to see the man or was the man carrying a telescope? Interestingly, if "man" is replaced by "picture", as shown in (2.9), the ambiguity disappears and only one reading remains possible. Our world knowledge implies that seeing an object using a telescope is quite normal, while what a picture can do with a telescope is unclear.

An example of lexical ambiguity is given in (2.10): *bank* can refer either to a financial institution or to a wall of a river channel. However, the following context makes us prefer the first reading, because we know that it is hardly possible to open accounts just sitting on a bank, but it is possible to do it in a financial institution.

(2.8) *John saw the man with a telescope*.

(2.9) *John saw the picture with a telescope.*

(2.10) *John went to the bank to open an account.*

*Bridging*

Bridging[21], or connecting parts of a discourse, implies a wide range of natural language inferences. One of the most studied bridging phenomena is anaphora. It is well known that syntactic and semantic agreement and parallelism are very helpful for anaphora resolution (see Mitkov, 1999). For example, given the sentences (2.11), in order to bind the anaphoric expression *John* to its antecedent *The boy* it is enough to find out that both expressions agree in number and belong to the same semantic typ, e.g., *human*. In order to resolve anaphora in (2.12) more information has to be involved: we need to know that the predicate *to be hungry* normally prefers a living being as its argument. Relating *house* and *door* in (2.13) presupposes knowledge about doors being parts of houses.

(2.11) *John reads a book. The boy likes reading.*

(2.12) *We gave the bananas to the monkeys because they were hungry.*[22]

(2.13) *John approached the house. The door was open.*

*Discourse Relations*

A discourse relation describes how two segments of discourse are logically connected to one another.[23] The sentences in Ex. (2.14) and (2.15) discussed by Lascarides and Asher (1993) have the same syntactic structure, but the corresponding events stand in different temporal relations. This follows from the background knowledge relating falling and pushing in a causative way, which is not the case for standing up and greeting. Similarly, reading (2.16) we understand that the alarm breaking event happened before waking up late and was the reason for it. We infer it using our world knowledge about typical waking up scenarios involving alarm bells.

(2.14) *Max stood up. John greeted him.*

(2.15) *John fell. Max pushed him.*

(2.16) *John woke up late today. His alarm broke.*

[21]See (Clark, 1975) for a classification of bridging types. A more recent overview of approaches to bridging is given in Asher and Lascarides (1998).

[22]This is a Wikipedia example.

[23]Discourse relations have been studied in (Hobbs, 1985a; Grosz and Sidner, 1986; Mann and Thompson, 1988; Lascarides and Asher, 1993) among others.

*Implicit Predicates*

When a predicate is highly predictable from the context, it can be omitted from the discourse.[24] Example (2.17) cannot be interpreted uniquely. But even if we do not know anything about John we can guess that he is either an author, an owner, an editor, or, for example, a seller of the mentioned book. This inference is possible because of our knowledge about the situations, in which a book (being an information container, a physical object, etc.) can be involved. Example (2.18) also lacks an explicit predicate. Knowing that the main feature of a wine is its taste, we can interpret (2.18) as *This wine is tasty*.

Noun compounds (2.19), possessives (2.20), and prepositional phrases (2.21) can be also interpreted in terms of implicit predicates. A morning coffee is most probably a coffee drunk in the morning, while a morning newspaper is a paper read in the morning. A Shakespeare's tragedy is a tragedy written by Shakespeare, while Shakespeare's house is a house where Shakespeare lives. *John in the house* describes the location of John, while *John in anger* denotes a state of John.

(2.17) *John finished the book.*

(2.18) *This wine is very good.*

(2.19) *morning coffee* vs. *morning newspaper*

(2.20) *Shakespeare's tragedy* vs. *Shakespeare's house*

(2.21) *John in the house* vs. *John in anger*

*Metaphor and Metonymy*

Rhetorical figures such as metaphor and metonymy are discourse phenomena requiring extremely strong reasoning capacities for their resolution.[25] Metaphor, or direct comparison of seemingly unrelated domains, requires deep knowledge about these domains, which allows us to find commonalities between them. Consider the famous citation from Shakespeare: "All the world's a stage". Later in the play the author explains which aspects of the concepts *world* and *stage* have to be compared ("And all the men and women merely players; They have their exits and their entrances"). Without this hint it would be difficult to come to a unique interpretation. Metaphors often do not presuppose a single reading leaving the reader with a spectrum of different associations (c.f. "Juliet is the sun" from

[24] See Pustejovsky (1991) for a detailed study of implicit predicates.

[25] See Fass (1997) for a detailed description of metaphor and metonymy.

Shakespeare). Because of the complexity and relatively low frequency of occurrence in texts, metaphors tend to be a rather marginal topic in NLP research.[26]

Metonymy, or using a word for a concept, which is associated with the concept originally denoted by the word, is usually easier to handle. In order to understand a metonymic expression one needs to know the associative link. For example, in the sentence (2.22) *White House* denotes not the building but the cabinet of the US president sitting in the building. Metonymy is closely connected to the notion of *regular polysemy*[27] which refers to a set of word senses that are related in systematic and predictable ways. Example (2.23) illustrates this phenomenon. In order to understand the proper relation between the referents of the word *school* in both sentences, we have to know that it can refer to different aspects of the concept *school*: building, institution, group of people, etc. This is a regular conceptual shift; the same set of senses is predictable for words denoting different types of organizations, e.g., firms, universities, ministries.

(2.22) *The White House supports the bill.*

(2.23) *John hurried up to the school. The school was going for an outing that day.*

## 2.4 Concluding Remarks

Computational natural language understanding implies automatically creating a formal representation of the text content. The more relevant information an NLU system manages to capture in this representation, the better it "understands" the text. The form and the content of the representation depend on the underlying theory of the linguistic meaning.

In linguistics, there are three main approaches to meaning: formal semantics, lexical semantics, and distributional semantics. These three frameworks, being for the most part orthogonal, consider different aspects of natural language semantics. Formal semantics is focused on the logical and compositional properties of language. Lexical semantics accounts for the organization of the lexical systems and semantic links between word senses. Distributional semantics regards properties of words as used in contexts. As for computational applications, formal semantic approaches mostly result in semantic parsers, while research in lexical and distributional semantics leads to construction of lexical-semantic databases (see Chap. 3).

[26]But see, for example, (Alonge and Castelli, 2003).
[27]This term was first introduced by Apresjan (1973).

In the study described in this book, we benefit from all the mentioned research directions. We use semantic parsers for producing logical representations of text fragments and enrich these representations on the basis of knowledge extracted from lexical-semantic databases. In addition, we use distributional information in order to recover those semantic relationships, which cannot be inferred with the help of lexical-semantic knowledge.

AI research on natural language understanding has been mostly focused on knowledge representation and reasoning techniques paying less attention to linguistic meaning. However, some of the recent AI approaches to NLU successfully employ sophisticated semantic parsers and lexical-semantic databases developed by computational linguists, see Chap. 4 for more details. Following this research direction, we use logical axioms for formalizing the developed integrative knowledge base and employ automated reasoning for drawing inferences relevant for natural language understanding.

This chapter briefly discusses the differences between linguistic knowledge and knowledge about the world. In this book, we do not make a distinction between lexical and world knowledge. We hope to have shown that a borderline between these two is difficult to draw, while non-linguistic knowledge about the world is crucial for interpretation of linguistic expressions.

## Chapter 3

# Sources of World Knowledge

In the area of artificial intelligence, interest to model world knowledge computationally arose in the late 1960s. The first proposals in this direction were Quillian's (1968) semantic networks and Minsky's (1975) frame-based representations. These proposals quickly attracted particular attention in the NLP community, because they seemed to provide a solution to the problems in natural language semantics, which required world knowledge. The two examples of the early classical approaches to natural language understanding employing semantic networks and frame representations are presented in (Woods *et al.*, 1980) and (Bobrow *et al.*, 1977).

The early approaches to knowledge modeling were quite informal and did not address the problem of defining a semantics for the knowledge representation languages as well as the problem of choosing the right conceptual entities for modeling the world. Later on, the importance of these issues was recognized and research on knowledge representation and reasoning split into two different paradigms.

The first line of research resulted in the construction of "clean" theory-based knowledge bases equipped with Tarski-style formalisms for characterizing their semantics.[1] Following the philosophical tradition, such knowledge bases have been called *ontologies*. The main purpose of ontologies is to enable efficient reasoning over their content and provide sufficient conceptual coverage of the modeled domain.

Researchers working in the second framework believe that a knowledge base appropriate for NLP should be based on words rather than on artificially created conceptual primitives. According to this approach, defining basic conceptual entities should mainly follow from corpus studies and psycholinguistic experiments. Initially, this view lead to the manual construction of electronic lexical-semantic dictionaries. Some of these dictio-

[1]Probably the most popular of the existing knowledge representation formalisms is called Description Logic (Baader *et al.*, 2003), see Sec. 4.4 for more details.

naries, especially WordNet (Fellbaum, 1998b), keep being extremely successful in the NLP community.

Starting from the 1990's, the quick progress of the statistical approaches to NLP has inspired many researchers working on lexical semantics to apply machine learning techniques to automatically extract lexical-semantic knowledge from corpora. This research direction is motivated by the belief that automating the knowledge acquisition process seems to be the only way to handle dynamic constantly changing knowledge, which builds the background of a language community.

As Poesio (2005) points out, the debates about what type of knowledge bases is more suitable for practical applications are still going on. Some researchers argue in favor of well-grounded manually developed ontologies, e.g., Smith (2004). Others think that knowledge underlying natural language understanding cannot and should not be logically grounded but should rather rely on evidence from psychology and corpora, e.g., Wilks (2002).

In this chapter, we will take a closer look at the two types of the knowledge bases employed in NLP: lexical-semantic resources and ontologies. We will outline the main features of the most popular resources providing world knowledge, compare them, and point out where these resources are redundant and which information relevant for natural language reasoning is still missing. In line with Poesio (2005), we hope to show that both lexical-semantic databases and ontologies have their relevance for the NLP research, especially what concerns natural language understanding.

## 3.1 Lexical-semantic Resources

Lexical-semantic resources contain information about the semantic organization of the lexicon of a particular language. Their developers try to capture regularities underlying use of words in the language. In the following, we consider the main aspects of lexical-semantic databases and review the most popular resources containing lexical-semantic knowledge, which have found their ways to the NLP community.

### *Lexical-semantic Relations*

Lexical-semantic relations are the central element in the organization of lexical-semantics knowledge bases.[2] These relations can be roughly divided into two classes: syntagmatic and paradigmatic. Words standing in a paradigmatic relation belong to the

[2]See Cruse (1986) for an overview and analysis of lexical-semantic relations.

same part of speech, share some features, and can substitute each other in some contexts, e.g., *dog* and *animal*. Syntagmatic relations are relations between words, which frequently co-occur in texts within a short span, e.g., *dog* and *bark*.

**Paradigmatic relations** The main types of paradigmatic relations are synonymy, hyperonymy, and meronymy. Lyons (1977) has defined *synonyms* as different words with almost identical or similar meanings, which are substitutable for each other without affecting the descriptive meaning of the utterances, e.g., *sick* and *ill*. Miller and Fellbaum (1991) have suggested a context sensitive definition of synonymy: “two expressions are synonymous in a context *C* if the substitution of one for the other in *C* does not change the truth value.” In the EuroWordNet project (Vossen, 2002), the following test is used to determine synonymy between *X* and *Y*: *A is (a/an) X* entails *A is (a/an) Y*, and vice versa.

X is a *hyperonym* of Y (or Y is a *hyponym* of X) if X is a generic term for Y and *A is X* entails but is not entailed by *A is Y*, e.g., *dog* and *animal*. The hyponym inherits all the features of the corresponding hypernym and adds at least one feature that distinguishes it from its superordinate and from other hyponyms of that superordinate. Hypernymy is often considered to be similar to conceptual inclusion, which is a transitive relation. The inheritance hierarchy based on hypernymy usually forms the backbone of most lexical-semantic resources.

*Meronymy* relates parts and wholes; this relation is often seen to be equal to the “part-of” relation in knowledge representation. Since there are different ways for something to be a part of something else, meronymy is considered to be a collection of relations rather than one single relation. For example, Winston *et al.* (1987) suggest six types of “part-of”: component–integral object (e.g., *leg–table*), member–collection (e.g., *soldier–army*), stuff–object (e.g., *alcohol–wine*), feature-activity (e.g., *pay–shop*), place–area (e.g., *oasis–desert*), portion–mass (e.g., *meter–kilometer*).

Another relation, which is included into some of the lexical-semantic databases, e.g., WordNet, is called *opposition*. If X is an opposite of Y then *A is X* entails *A is not Y* and vice versa, e.g., *short–long*, *male–female*, *off–on*. The opposition relation does not imply a maximum degree of difference in meaning between two words. Rather, the opposite words must be similar in all respects but one. The term *antonym* has been commonly used as equal to *opposite*; however, sometimes it is restricted to the gradable opposites only (e.g., *short–long*), see Lyons (1968); Cruse (1986), whereas for all other types of opposition the general term *opposite* is used.

As the test patterns described above suggest, hypernymy and meronymy are mostly applicable to nouns, while oppositions are mostly defined for adjectives. Semantic relations for verbs are much less standardized; every resource introduces its own list of relations. For example, WordNet (Fellbaum, 1998b) includes opposition (e.g., *give–take*) and variants of entailment such as causation (e.g., *show–see*), backward presupposition (e.g., *forget–know*), and temporal inclusion subdivided into proper inclusion (e.g., *walk–step*) and coextensiveness (e.g., *march–walk*).

**Syntagmatic relations** Word combinations related with syntagmatic relations are usually classified according to the word association strength.[3]

- *Idioms* are rigid word combinations, which are comprehended as a whole; its meaning cannot be determined from the meaning of its parts, e.g., *to kick the bucket*.
- *Collocations* (or co-occurrences) are combinations of words, which co-occur in text spans of a defined size more often than it would be expected by chance, e.g., *dog–bark*.[4]
- *Free word combinations* are combinations of independent words.

In computational semantics, syntagmatic relations as considered to form a continuum “spanning the range from free-combining words to idioms“ (Viegas *et al.*, 1998). In lexical-semantic databases, idioms are usually included as one single element; for example, *to kick the bucket* is represented by one lexical node both in WordNet (Fellbaum, 1998a) and in FrameNet (Ruppenhofer *et al.*, 2010). Collocations are represented as relations between different lexical items.

In contrast to most of the paradigmatic relations, collocations are probabilistic rather than deterministic in character. For example, the syntagmatic relation between *dog* and *bark* does not imply that any mentioning of a dog in a text is accompanied by an implicit barking event. In contrast, the paradigmatic relation *dog–animal* should imply that any entity being referred to as “dog” can be referred to as “animal”. This fact prevents some lexicographers from including syntagmatic relations into lexical-semantic databases, which are not designed for accommodating probabilistic information. At the same time, collocations are successfully employed in statistical NLP (cf. Sec. 2.2.1).

[3]See Viegas *et al.* (1998) for a discussion of the role of syntagmatic relations in computational linguistics.

[4]Some researchers understand the term *collocation* in a narrower sense, namely, as referring to word combinations frequently co-occurring in syntactic phrases, e.g., the *dog–bark* relation falls under this definition, but not *doctor–hospital*.

*Reasoning with Lexical-semantic Relations*

Although lexical-semantic databases are usually not designed for reasoning, such relations as hypernymy or meronymy are often considered to imply inferences. As already mentioned, a hyponym is supposed to inherit all features of its hypernym. For example, *animal* is defined in WordNet as “a living organism characterized by voluntary movement”. This definition is inherited by *dog*, *cat*, *bird*, etc., which are hyponyms of *animal*. However, such inheritance does not always work correctly. For example, in WordNet, *chair* is a hyponym of *furniture* and *wheelchair* is a hyponym of *chair*. Since hypernymy is a transitive relation, *wheelchair* should inherit the definition of *furniture*, which is “furnishings that make a room or other area ready for occupancy”. There is a contradiction, because a wheelchair is not intended for furnishing a room.

This problems are not specific to the lexical system, but rather concern organization of the common-sense knowledge in general. Experimental findings summarized in Prototype theory (see Sec. 2.2.1) suggest that the common-sense knowledge is structured around prototypes, which may serve as a basis for concept definitions. At the same time, borderline category members (like *wheelchair*), which do not completely satisfy the category definitions seem to be perfectly acceptable in the human cognitive system.

Another problem in organization of lexical hierarchies concerns structural inconsistencies. For example, WordNet defines the synset $S_1 = \{$*inhibit*, *bottle up*, *suppress*$\}$ to be a hypernym of the synset $S_2 = \{$*restrain*, *keep*, *keep back*, *hold back*$\}$. At the same time, $S_2$ is a hypernym of $S_1$.[5] Apart from the fact that definition circles are conceptually wrong, such cycles can be problematic with respect to reasoning, because they can provoke infinite reasoning loops.

*Word Senses*

Lexical-semantic relations are usually defined on *word senses* rather than on words. A word sense is one of the meanings of a word. The capacity of a word to have multiple meanings is called *polysemy*. For example, in WordNet (Fellbaum, 1998b), the word *play* currently has 17 senses as a noun and 35 senses as a verb. Sense discrimination is usually based on differences in contexts of the word usage in text. For example, the sentences *Shakespeare wrote a play* and *The ball is still in play* imply different senses of *play*. In

[5]This example is taken from Richens (2008). This paper also discusses other types of structural problems in WordNet-like lexical-semantic databases.

WordNet, the first sense is a hyponym of *dramatic composition*, while the second sense is a hyponym of *activity*.

For dictionary developers, it is often difficult to decide whether a particular sense distinction is reasonable or not. For example, does the verb *play* have the same or different meanings in the sentences *The tape was playing for hours* and *The band played all night long*?[6] Several tests have been proposed for determining whether a sense discrimination is necessary (see Cruse, 1986, for an overview). For example, one approach relies on detecting whether two occurrences of a word form imply different semantic relations with other items. One of the criteria formulated by Cruse (1986) is the following:

> If there exist a synonym of one occurrence of a word form, which is not a synonym of a second, syntactically identical occurrence of the same word form in a different context, then that word form is ambiguous, and the two occurrences exemplify different word senses.

For example, given two sentences *Guy struck the match* and *The match was a draw*, one might suggest *lucifer* as a synonym for *match* in the first sentence (but not in the second) and *contest* as a synonym in the second sentence (but not in the first). Cruse (1986) concludes that *match* is ambiguous and these two sentences represent its different senses.

Besides theory-based approaches to word sense discrimination, which usually underly manual construction of lexical-semantic databases, there are approaches to automatic learning of word senses from corpora. These approaches are based on the distributional hypothesis (see Sec. 2.2.1) representing the intuition that words that occur in the same contexts tend to be semantically similar. Word sense discrimination algorithms automatically discover word senses by clustering words according to their distributional similarity. Each cluster that a word belongs to corresponds to a sense of the word. For example, the *clothing* sense of the word *suite* can be represented by the cluster {*blouse*, *slack*, *legging*, *sweater*} and the *litigation* sense – by the cluster {*lawsuit*, *allegation*, *case*, *charge*}.[7] A brief overview of automatic methods for word sense discrimination is given, for example, by Pantel and Lin (2002).

In contemporary lexical semantics, the notion of word sense is still quite problematic, and there are researchers, which question its principle relevance for theoretical linguistics and natural language processing (for example, see Kilgarriff, 1997).[8] Nevertheless, most

[6] The examples are taken from WordNet 3.0.

[7] This example is taken from Pantel and Lin (2002).

[8] The study of polysemy and the criteria for distinguishing word senses is one of the most active areas in lexical semantics. Since the detailed discussion of polysemy is not directly relevant for this book, we would like to refer

of the existing lexical-semantic resources rely on word sense distinctions for organization of lexical-semantic knowledge.

*Lexemes vs. Concepts*

Strictly speaking, lexical-semantic databases provide information about words and not about the world. However, this distinction is difficult to make. Obviously, our conceptualization of the world is somehow reflected in the organization of the lexicon. But how much lexical and conceptual systems do overlap and whether there is a difference between them is still an open question. Some researchers believe that word senses actually *are* concepts, i.e. cognitive units of knowledge.[9] This approach was taken, for example, by the researchers working in the framework of cognitive semantics (Tarnawsky, 1982; Johnson-Laird, 1983; Jackendoff, 1983). Jackendoff (1983) claims that there is no distinction between the semantic and conceptual levels:

> word meanings are expressions of conceptual structure. That is, there is not a form of mental representation devoted to a strictly semantic level of word meaning, distinct from the level, at which linguistic and nonlinguistic information are compatible. This means that if, as it is often claimed, a distinction exists between dictionary and encyclopedic lexical information, it is not a distinction of level; these kinds of information are cut from the same cloth.

Other researchers claim that there are features of lexical systems that do not correspond to conceptualizations and some useful conceptual distinctions are not reflected in language. For example, Hirst (2004) lists three points distinguishing the lexical and conceptual systems. First, according to Hirst, natural language lexicons are sometimes redundant with respect to conceptual organization of knowledge in the sense that synonyms and near-synonyms expressing very fine-grained meaning distinctions (if any), e.g., *sick* and *ill* probably refer to the same concept. Second, there are concepts, which are not lexicalized and require a multi-word description in order to be referred to in the language. For example, Dutch has no words corresponding to the English words *container* or *coy*; Spanish has no word corresponding to the English verb *to stab* etc. Apart from lexical gaps in one language relative to another, new concepts, which have not been lexicalized yet in any language constantly arise. Hirst's third point concerns linguistic categorizations, which are not relevant for conceptual knowledge organization. For example, some languages distinguish between objects that are discrete and those that are not: countable and mass nouns.

the interested reader to the Leacock and Ravin's book on this topic (Leacock and Ravin, 2000), which gives an exhaustive overview of the modern linguistic approaches to polysemy.

[9]Philosophical discussion about the nature and structure of concepts is out of the scope of this book. For a brief overview of the philosophical theories of concepts, see Margolis and Laurence (2005).

This is an important conceptual distinction; however, in practice the actual linguistic categorization might be rather arbitrary. For example, *spaghetti* is a mass noun, but *noodle* is countable in English.[10]

Some researchers question the principle idea of viewing lexical-semantic relations as expressing conceptual relationships (e.g., Lenci *et al.*, 2006; Murphy, 2010). For example, the acceptability of both *A is a performer* $\rightarrow$ *A is a person* and *A is a dog* $\rightarrow$ *A is an animal* (at least for a prototypical case) allows us to establish the hypernymy relation between *performer* and *person*, as well as the relation between *dog* and *animal*. However, the conceptual relation linking *performer* and *person* is crucially different from the relation holding between *dog* and *animal*. Being an animal as an intrinsic property of every dog, while for a person to be a performer is a temporal role related to a specific activity. Lenci *et al.* (2006) conclude that "linguistic tests [...] undertermine the real type of conceptual relations that link two concepts".

Lexical-semantic resources can be subdivided into hand-crafted electronic dictionaries and automatically generated databases. Although the developers of both types of resources may strive for similar goals, they use quite different methods and happen to arrive at different results. The mentioned division underlies the organization of the following subsections.

### 3.1.1 *Hand-crafted Electronic Dictionaries*

In electronic dictionaries, every word is linked to a set of word senses, which are united into groups of semantically similar senses. Different types of semantic relations are then defined on such groups, e.g., taxonomic, part-whole, or causation relations. The resources considered in this section have been created manually on the basis of corpus annotation, psycholinguistic experiments, and dictionary comparison.

#### 3.1.1.1 *WordNet Family*

Lexical-semantic resources belonging to the WordNet Family[11] have the longest and the most successful history in the NLP applications because of their large lexical coverage, variety of instantiated semantic relations, and transparent organization of the content. Originally, English WordNet[12] was created under the direction of George Miller (Miller *et al.*,

[10] Similar ideas are expressed by Nirenburg and Raskin (2004).
[11] *http://www.globalwordnet.org/*
[12] *http://wordnet.princeton.edu/*

1990). Because of its enormous success, for many other languages versions of WordNet based on the same principles have appeared.

In WordNet, lexical-semantic knowledge is represented in a network-like structure. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms called *synsets*. Synsets actually contain not words, but word senses. For example, the word *board* participates in several synsets ({*board*, *plank*}, {*board*, *table*}, etc.) which refer to its different senses. In the semantic community, synsets are often considered to be equivalent to concepts. Synsets and single lexemes are interlinked by means of relations including classical lexical-semantic relations: hyponymy, meronymy, antonymy, etc. For example, the synset {*animal*,...} is defined in WordNet as a hypernym of the synset {*dog*,...}. The statistics for English WordNet version 1.3 is presented in Table 3.1.

At present, WordNet contains dense webs of semantic relations defined between synsets that belong to the same part-of-speech, while the number of relations between different parts of speech is relatively small. Thus, there exist not one, but four almost separated WordNets: WordNet for nouns, for verbs, for adjectives, and for adverbs. The part of speech distinction as a basis for building synsets makes WordNet linguistically, but not cognitively motivated.

The only semantic relation linking synsets from different part-of-speech hierarchies is the attribute-value relation (cf. Table 3.1). The derivational relation links specific word senses of lexemes, which are morphologically similar, e.g., *construct-construction*. Unfortunately, WordNet does not define the type of the semantic relation implied by morphological similarity. There exists an attempt to annotate WordNet derivational pairs with semantic relations.[13] This has been done for 17740 verb-noun pairs. Such associations like, for example, *dog-bark* or *water-wet* are currently not represented in WordNet.

Concerning its practical applications in reasoning, WordNet has two main shortcomings. First, some researchers criticize its word sense distinctions for being unnecessarily fine-grained which complicates automatically distinguishing between word senses (see, for example, Agirre and Lacalle, 2003). Second, WordNet does not guarantee any kind of conceptual consistency, which may imply incorrect reasoning, compare Oltramari *et al.* (2002) who present a methodology for cleaning up the hypero-hyponymical hierarchy in the top-level of WordNet. Restructuring of the base level of WordNet in order to guarantee its consistency still needs further investigation. Moreover, other types of WordNet relations besides hyperonymy need to be cleaned up as well.

[13] *http://wordnet.princeton.edu/wordnet/download/standoff/*

Table 3.1 English WordNet 1.3 statistics

| **POS** | **Lexemes** | **Synsets** |
|---|---|---|
| Nouns | 117798 | 82115 |
| Verbs | 11529 | 13767 |
| Adjectives | 21479 | 18156 |
| Adverbs | 4481 | 3621 |

| **Relation** | **Total** | **Defined on** | **POS** | **Example** |
|---|---|---|---|---|
| Hyperonymy | 89089 | synsets | same POS | {*dog*,...} - {*animal*,...} |
| Instantiation | 8577 | synsets | nouns | {*Lennon*,...} - {*rock star*,...} |
| Member meronymy | 12293 | synsets | nouns | {*policeman*,...} - {*police*,...} |
| Part-of meronymy | 9097 | synsets | nouns | {*leg*,...} - {*table*,...} |
| Substance meronymy | 797 | synsets | nouns | {*tissue*,...} - {*organism*,...} |
| Domain category | 9500 | synsets | undef. | {*fly ball*,...} - {*baseball*,...} |
| Entailment | 408 | synsets | verbs | {*license*,...} - {*approve*,...} |
| Causation | 220 | synsets | verbs | {*kill*,...} - {*die*,...} |
| Similarity | 21386 | synsets | adj. | {*full-length*,...} - {*whole*,...} |
| Antonymy | 8689 | word senses | same POS | *cold - hot* |
| Attribute-value | 1278 | synsets | noun-adj. | {*complexity*,...} - {*complex*,...} |
| Derivational | 74718 | word senses | verb-noun | *construct - construction* |

Nevertheless, as already mentioned, because of its huge lexical and conceptual coverage, WordNet has been extremely popular and widely used in different NLP applications (see Morato *et al.*, 2004, for an overview).

#### 3.1.1.2 *Resources for Semantic Role Labeling*

Lexical-semantic resources for semantic role labeling (SRL) provide a basis for defining semantic relations between predicates and their arguments. For example, in the sentence *John gave Mary a book* the verb *give* has three arguments: *John*, *Mary*, and *book*. The arguments are semantically related to the verb predicate; this relation can be captured in terms of semantic roles. Thus, the argument *John* might be annotated with *Donor*, *Agent*, or *Arg0* depending on the underlying linguistic theory. Since the notion of semantic roles first appeared in the literature introduced in generative grammar during the mid-1960s and early 1970s (cf. Sec. 2.2.1), the debates about the exact set of roles appropriate for representing argument structures keep going on. In the following, we describe and compare three main resources for SRL, which suggest different solutions for the problem of choosing roles.

*Propositional Bank*

The original goal of the Propositional Bank (PropBank) project (Palmer *et al.*, 2005) was to add a layer of semantic role labeling to the syntactic annotation of the Penn Treebank (Marcus *et al.*, 1994) and to provide consistent descriptions for different syntactic realizations of predicate-argument structures. PropBank does not define a universal set of semantic roles, which are rather considered to be verb-specific. Semantic arguments of every verb are numbered. *Arg0* generally stands for a prototypical agent and *Arg1* is a prototypical patient or theme (cf. Dowty (1991) for description of prototypical roles) as, for example, in $[John]_{Arg0}$ *broke* $[the\ window]_{Arg1}$ and $[The\ window]_{Arg1}$ *broke*. Arguments with higher numbers allow no consistent generalization.

PropBank contains a lexicon entry for every verb occurring in the Penn Treebank. Senses of polysemous verbs correspond to *framesets*. A separate set of numbered roles, a *roleset*, is defined for each frameset. A verb's *frame* is then the collection of the frameset entries for the verb. For example, the frame for the verb *break* contains 9 different framesets; the first one has 4 roles: *Arg0* breaker, *Arg1* thing broken, *Arg2* instrument, *Arg3* pieces.

Since the role descriptions are given in PropBank in natural language form, the resource is practically useless for reasoning. The only useful information it provides concerns mapping the verb arguments to the appropriate slots in the framesets.

PropBank annotates around 1 million words from the Wall Street Journal including 3 633 unique verbs.

*VerbNet*

The English verb lexicon VerbNet[14] (Kipper *et al.*, 2000) consists of verb classes, which are based on Levin's classification of verbs (Levin, 1993). The motivation behind designing such a resource is to provide the link between syntax and semantics of verbs.

[14] *http://verbs.colorado.edu/ mpalmer/projects/verbnet.html*

Each VerbNet class contains a set of verbs and is characterized by a list of arguments similar of the classical thematic roles (*Agent*, *Theme*, *Location*, etc.). These roles have semantic restrictions, which are organized in a small hierarchy (around 40 nodes). For example, for the class `hit-18.1` the following roles with restrictions are introduced: *Agent*[+intentional_control], *Patient*[+concrete], *Instrument*[+concrete].

Several syntactic frames are assigned to every verb class. Syntactic frames describe possible surface realizations for the verbs in the class. A syntactic frame consists of the verb itself, the thematic roles in their preferred syntactic positions around the verb, and other lexical items, which may be required for a particular construction or alternation. For example, for the class `hit-18.1` the following syntactic frames are defined: *Agent V Patient* (*John hit the ball*), *Agent V at Patient* (*John hit at the window*), *Agent V Patient[+plural] together* (*John hit the sticks together*). The semantics of a syntactic frame is captured through a conjunction of semantic predicates. For example, the *Agent V Patient* frame in the `hit-18.1` class has the following semantics: *cause(Agent, Event)*, *manner(during(Event),directed_motion,Agent)*, *contact(end(Event), Agent, Patient)*, etc.

At the moment semantic predicates in VerbNet as well as its semantic restrictions are just labels; they are not axiomatized or linked to any formal theory. This makes this information useless for reasoning. Similar to PropBank, the only applications of VerbNet in NLP concern usage of its syntactic patterns for mapping verb arguments to appropriate roles. The generalization, which VerbNet gives as compared to PropBank is the unification of syntactic frames into verb classes, which might be seen as referring to conceptual entities.

VerbNet version 3.0 contains 3769 verbs, 274 verb classes, and 23 thematic roles.

*FrameNet*

FrameNet[15] (Ruppenhofer *et al.*, 2010) is based on Fillmore's frame semantics (Fillmore, 1976) and supported by corpus evidence. The aim of the FrameNet project is "to document the range of semantic and syntactic combinatoric possibilities (valences) of each word in each of its senses" (Ruppenhofer *et al.*, 2010). In contrast to the SRL resources described above, FrameNet assigns semantic roles not only to verbs, but also to nouns, adjectives, adverbs, and prepositions. Moreover, lexemes belonging to different parts of speech (like *construct* and *construction*) can be assigned to the same frame.

[15] *http://framenet.icsi.berkeley.edu/*

The lexical meaning of predicates in FrameNet is expressed in terms of frames, which are supposed to describe prototypical situations spoken about in natural language. Every frame contains a set of roles corresponding to the participants of the described situation, e.g., DONOR, RECIPIENT, THEME for the GIVING frame. Predicates with similar semantics evoke the same frame, e.g., *give* and *hand over* both refer to the GIVING frame. For example, given the phrase *John gave Mary a book*, *John* is annotated with the DONOR role, *Mary* is annotated with RECIPIENT, and *book* is annotated with THEME. The FrameNet semantic roles are more specific than in the previously described resources and often refer to concrete scenarios, e.g., DONOR instead of *Agent*.

FrameNet provides information about syntactic realization patterns of frame elements. For example, the role RECIPIENT in the frame GIVING is most frequently filled by a noun phrase in the indirect object position or by a prepositional phrase with the preposition *to* as the head in the complement position.

In addition, FrameNet introduces semantic relations, which are defined on frames. For example, the GIVING and GETTING frames are connected by the causation relation. Roles of the connected frames are also linked, e.g., DONOR in GIVING is linked to SOURCE in GETTING. This feature of FrameNet makes this resource highly useful for reasoning. Frame relations suggest themselves to be used for detection of paraphrases like $[John]_{\text{DONOR}}$ $[gave]_{\text{GIVING}}$ $[Mary]_{\text{RECIPIENT}}$ $[the\ book]_{\text{THEME}}$ and $[Mary]_{\text{RECIPIENT}}$ $[got]_{\text{GETTING}}$ $[the\ book]_{\text{THEME}}$ $[from\ John]_{\text{DONOR}}$.

Thus, FrameNet positively differs from the resources described above in the following ways: a) frames are not part-of-speech specific, which makes it possible to unify lexemes from different categories referring to the same concept, b) semantic relations defined on frames also link frame roles, which opens a whole range of new reasoning options. Nevertheless, FrameNet has shortcomings, which should be carefully taken into account by researchers willing to apply it for inferencing. These shortcomings are discussed in Sec. 6.1.

English FrameNet version 1.5 statistics is given in Table 3.2.[16]

FrameNet has a shorter history in NLP applications than WordNet, but lately more and more researchers demonstrate its potential to improve the quality of question answering (e.g., Shen and Lapata, 2007) and recognizing textual entailment (e.g., Burchardt *et al.*, 2009).

[16]For more information about each frame relation see Sec. 5.2.2.

Table 3.2 English FrameNet 1.5 statistics.

| **Entity** | **Total** |
|---|---|
| Verbs | 4605 |
| Nouns | 4742 |
| Adjectives | 2122 |
| Adverbs | 167 |
| Prepositions | 143 |
| Frames | 1019 |
| Frame roles | 8884 |

| **Frame relation** | **Total** | **Example** |
|---|---|---|
| Inheritance | 617 | GIVING - COMMERSE_SELL |
| Causative_of | 48 | KILLING - DEATH |
| Inchoative_of | 16 | COMING_TO_BE - EXISTENCE |
| Perspective | 99 | OPERATE_VEHICLE - USE_VEHICLE |
| Precedence | 79 | FALL_ASLEEP - SLEEP |
| Subframe | 117 | SENTENCING - CRIMINAL_PROCESS |
| Using | 490 | OPERATE_VEHICLE - MOTION |
| See_alo | 41 | LIGHT_MOVEMENT - LOCATION_OF_LIGHT |

#### 3.1.1.3 *WordNet vs. FrameNet*

WordNet and FrameNet are currently the semantically richest lexical-semantic dictionaries available. WordNet has a huge lexical coverage and a fine-grained hierarchy of word senses especially elaborated for nouns.[17] FrameNet looks deeper into the structure of the analyzed conceptual entities introducing frames defined through role sets. Moreover, instead of documenting all possible nuances of word senses, FrameNet generalizes them into relatively abstract, but still specific enough scenarios (frames), which constitute a network of a limited size (only about 1 000 frames against 118 000 synsets). An aspect, in which FrameNet significantly yields to WordNet, is lexical coverage (12 000 FrameNet lexical entries against WordNet's 155 000). This problem of FrameNet is well-known in the NLP community and some researchers have tried to extend its lexicon with the help of

[17] As stated above, some researchers see the level of granularity, which WordNet provides, as too detailed for being useful in practice.

WordNet relations (Burchardt *et al.*, 2005) or corpus-based techniques (Cao *et al.*, 2008). Unfortunately, both of these approaches introduce noise, which reduces the quality of the hand-crafted resource.

Anyway, FrameNet, even with an extended lexicon, could not fully substitute WordNet, because the former completely lacks many of the useful semantic relations such as meronymy, instantiation, etc. introduced in the latter. Although there is some informational overlap between WordNet and FrameNet, these resources seem to be in large part disjoint. WordNet's strong part is the relations between noun synsets, which refer to objects, whereas FrameNet specializes on relations defined on events. Unfortunately, a direct link between FrameNet and WordNet does not exist yet, although one could take a detour via VerbNet. The VerbNet research group has created a mapping between VerbNet and WordNet, VerbNet and PropBank, and VerbNet and FrameNet, see the SemLink project webpage.[18] At the moment, the SemLink mapping contains only verbs, which are included in VerbNet.

One important type of lexical-semantic relations has not been covered neither by WordNet nor by FrameNet yet. It concerns typical relations between predicates and their arguments, which are represented by selectional preferences and qualia structures (see Sec. 2.2.1). As for selectional preferences, there have been many studies on automatic extraction of this information from corpora (for more details see Sec. 3.1.2). Concerning qualia structures, there exists a manually created resource for English based on the Generative Lexicon, which is called the Brandeis Semantic Ontology (BSO; Pustejovsky *et al.*, 2006). Unfortunately, BSO has not been released yet[19] and there is no sufficient documentation of the resource. Experiments on automatic extraction of qualia structures from corpora are mentioned in Sec. 2.2.1.

### 3.1.2 *Automatically Generated Lexical-semantic Databases*

In this section, we document some of the successful attempts of automatic learning of lexical-semantic knowledge. All mentioned approaches employ methods and algorithms developed in the framework of distributional semantics (see Sec. 2.2.1).

[18] *http://verbs.colorado.edu/semlink/*
[19] See the online interface at *http://eurydice.cs.brandeis.edu/BSOonline/newBSO/BSObrowser.py*.

#### 3.1.2.1 *Learning Semantic Relations*

The largest part of the research on automatic learning of semantic relations has been dedicated to noun-noun relations.

One approach focusing on interpretation of noun compounds is based on paraphrasing via prepositional phrases (Lauer, 1995; Lapata and Keller, 2004), for example, *paper bag* → *bag for papers* (purpose), *paper bag* → *bag of paper* (material/part-of).[20]

Another approach to learning noun-noun relations employs lexico-syntactic patterns (Hearst, 1992, 1998; Girju *et al.*, 2006). For example, the pattern $\texttt{NP}_0$ `such as` $\texttt{NP}_1$`,` $\texttt{NP}_2$`, ..` $\texttt{NP}_n$ can be used for detecting the hypernym relation in the text fragments like *domestic animals such as cats, dogs, and parrots*.

More recent systems performing learning of noun-noun relations are described in the proceedings of SemEval 2007 that included a special competition track on "Classification of Semantic Relations between Nominals" (Girju *et al.*, 2007). In practical NLP, automatically learned semantic relations have been applied in question answering (Girju, 2001; Girju *et al.*, 2003).

#### 3.1.2.2 *Learning Selectional Preferences*

Most of the approaches to automatic learning of selectional preferences rely on corpus-based frequencies of the predicate-argument constructions, see Schulte im Walde (2010) for an overview. For every predicate in the corpus, words occurring as arguments of this predicate are collected and generalized as WordNet synsets (Resnik, 1997; Abney and Light, 1999; Clark and Weir, 2002) or distributional clusters (Pereira *et al.*, 1993; Rooth *et al.*, 1999). These generalizations can be further used to predict selectional preferences for unseen arguments (Erk, 2007).

In NLP, selectional preferences have been employed for word sense disambiguation (Carroll and McCarthy, 2000), detecting metonymy (Nastase and Strube, 2009), and for refining automatically learned entailment rules (Basili *et al.*, 2007; Pantel *et al.*, 2007).

#### 3.1.2.3 *Learning Entailment Rules*

The DIRT (Discovery of Inference Rules from Text) database is a collection of paraphrases automatically learned from corpora (over a 1 GB set of newspaper text), see Lin and Pantel (2001). The DIRT methodology presupposes that a path, extracted from a dependency parse tree, is an expression that represents a binary relationship between two

[20] This example is provided by Girju (2009).

Table 3.3 VerbOcean statistics.

| **Relation** | **Total** | **Example** |
|---|---|---|
| Similarity | 11515 | *X produce Y - X create Y* |
| Strength | 4220 | *X wound Y - X kill Y* |
| Antonymy | 1973 | *X open Y - X close Y* |
| Enablement | 393 | *X fight Y - X win Y* |
| Happens-before | 4205 | *X marry Y - X divorce Y* |

nouns. Thus, if two paths tend to link the same sets of words then the meanings of the corresponding patterns are similar. For example, the patterns *X wrote Y* and *X is the author of Y* are similar according to DIRT. For every two similar patterns, DIRT provides a measure of their similarity. DIRT contains around 231 000 unique patterns.

VerbOcean[21] is a semantic network of verbs containing 3 477 unique lexemes (Chklovski and Pantel, 2004). In this approach, semantic relations are contracted by querying the Web with Google for lexico-syntactic patterns indicative of each relation. For example, *Xed * by Ying the* is a pattern for the enablement relation. After the pairs of semantically related verbs had been found, the DIRT methodology was applied for extracting corresponding paraphrases. The result is a database containing paraphrases annotated with semantic relations, e.g., *X outrage Y* happens-after/is stronger than *X shock Y*. The statistics of the VerbOcean semantic relations is presented in Table 3.3.

The WikiRules! resource[22] contains about 8 million lexical reference rules extracted from Wikipedia as described in (Shnarch *et al.*, 2009). A lexical inference rule *LHS* ⇒ *RHS* indicates that the left-hand-side term refers to a possible meaning of the right-hand-side term, for example, *Margaret Thatcher* ⇒ *United Kingdom*. In the WikiRules! resource, a title of a Wikipedia article is taken as *LHS* of the constructed rule, while a definition term extracted from the first sentence of the article is taken as *RHS*. The rules extracted by this method mostly involve named entities and terminological terms, typically covered in encyclopedias.

[21] *http://demo.patrickpantel.com/demos/verbocean/*
[22] *http://u.cs.biu.ac.il/ nlp/downloads/WikiRules.html*

Naturally, databases containing entailment rules have been employed in recognizing textual entailment[23]; for a list of systems using DIRT, VerbOcean, and WikiRules!, see the web portal *RTE Knowledge Resources*.[24]

## 3.2 Ontologies

*What is an Ontology?*

Historically, the notion of *ontology* comes from metaphysics – a branch of philosophy. In the philosophical sense, ontology is the study of the nature of being, existence or reality in general, which tries to define the basic fundamental categories of being. In the 1970s, researchers working in the field of artificial intelligence realized that capturing world knowledge in the form of a database is one of the keys to building adequate AI systems. Following the philosophical tradition, AI researchers labeled the developed computational models of the world as ontologies. Nowadays, the term "ontology" in the information science community refers both to the theory about how to model the world and to the specific engineering artifacts. In this book, we will use the term in the latter sense.

In information science, the most famous definition of ontology was suggested by Gruber (1993) who claims that "an ontology is an explicit specification of a conceptualization". Guarino (1998) has elaborated on this definition developing a formal intensional account of the notion of ontology:

> An ontology is a logical theory[25] accounting for the *intended meaning* of a formal vocabulary, i.e. its *ontological commitment* to a particular *conceptualization* of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models.

Guarino illustrates his definition by a diagram, which is replicated in Fig. 3.1. In Guarino's view, a conceptualization is a language-independent view of the world (a set of conceptual relations defined on a domain space), while an ontology is a language-dependent cognitive artifact, which is committed to a certain conceptualization of the world by means of a given language.[26]

[23] See Sec. 8.1.1 for a description of the recognizing textual entailment task.
[24] *http://www.aclweb.org/aclwiki/index.php?title=RTE_Knowledge_Resources*
[25] By "logical theory" Guarino means a set of axioms.
[26] By "language" Guarino means an ontology representation language rather than a natural language.

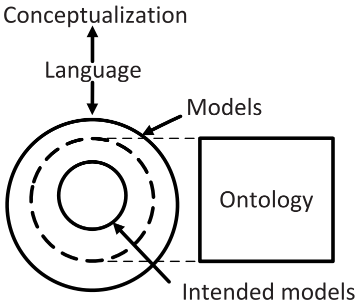


Fig. 3.1 Illustration of the definition for the term "ontology" given by Guarino (1998).

*Ontology Content*

According to the Guarino's definition, an ontology provides a set of axioms, which formally constrain the intended meaning of a vocabulary. Thus, for Guarino, an ontology defines a *terminology* of a language and contains axioms like $\forall x\ (\textsc{Pacific\_island}(x) \rightarrow \textsc{Island}(x) \wedge \exists y\ (\textsc{locatedIn}(x,y) \wedge \textsc{Pacific\_ocean}(y)))$, which axiomatizes the term "Pacific island" by defining it as an island located in the Pacific ocean. In applied information science, computational models of the world called ontologies are not limited to terminological knowledge (definitions of classes); they can also include factual knowledge about individuals as well as complex theories defining relationships between different classes. For example, given a class PACIFIC_ISLAND, an ontology can introduce *Tahiti* as an individual instantiating this class.

Thus, in practice, an ontology can include any type of world knowledge represented by axioms. The main atomic entities constituting ontological axioms are classes, relations, and individuals. In the "Pacific ocean" example above, PACIFIC_OCEAN, ISLAND, and PACIFIC_OCEAN are classes and LOCATEDIN is a relation. The main purpose of ontologies is to enable inferences over their content. For example, if we define *Tahiti* as an instance of the class PACIFIC_ISLAND then we can infer that Tahiti is located in the Pacific ocean.

*Ontology Representation*

In order to enable reasoning, an ontology must be specified using some concrete formal representation. Modern ontology representation languages are based on logical formalisms, which are usually a variant of the first-order predicate calculus. There exists a variety of languages, which can be used for representation of ontologies (see de Bruijn, 2003, for an overview), with varying characteristics in terms of their *expressiveness* and *computational complexity*.

The expressive power of a representation language depends on the range of constructs that can be used to describe the components of an ontology, i.e. it characterizes the syntactic richness of the language. Computational complexity of a language is determined by the complexity of such problems as, for example, checking whether a set of axioms formulated in this language is logically valid.[27] There is an obvious trade-off between expressivity and complexity. As languages become more and more expressive, the computational complexity of reasoning increases. The language chosen for ontology representation must be able to express all the desired concepts in the domain of interest. However, the language must also support efficient reasoning, especially if it is designed for representing large knowledge bases.

Another important characteristics of an ontology representation language is *decidability* of the underlying logic. A logical system is decidable if there is a method for determining whether arbitrary formulas are logically valid in this system.[28] If an ontology is represented using a formal language based on an undecidable logic, then reasoning with this ontology can fail for some queries. For example, first-order logic (FOL) and its extensions are not decidable.

### *Background Theory for Ontology Modeling*

Ontologies are intended to represent one particular view of the modeled domain (rather than a diversity of views and term usages) in an unambiguous and well-defined way. In contrast to lexical-semantic resources, ontologies do not allow usage of the same term in different senses and are supposed to provide clean and precise term definitions. Thus, ontologies as domain models are much closer to "scientific" theories than to fuzzy common sense knowledge reflected in the organization of natural language lexicons. This feature of ontologies influences ontology representation languages, which usually do not tolerate inconsistencies and ambiguities and leave no space for probabilistic knowledge.

[27] For the notion of computational complexity, see, for example, (Papadimitriou, 1994).
[28] A detailed consideration of the issues concerning decidability and logical validity can be found, for example, in (Monk, 1994).

*Interfacing with Lexicons*

Ontological vocabularies consist of artificially created concept labels rather than words. Therefore for an ontology to be used by an NLP application, it is necessary to have an interface to a natural language lexicon. Prevot *et al.* (2005) list the following three methods of interfacing ontologies and lexical resources.

- *Restructuring* a computational lexicon on the basis of ontological-driven principles
  This option concerns only the lexical resource, whereas the ontology cannot be used in the actual application.
- *Populating* an ontology with lexical information
  This option presupposes mapping lexical units to ontological primitives. In this simplifying view, a computational lexicon is considered to contain lexicalized concepts, which should be mapped to the corresponding concepts in the ontology.
- *Aligning* an ontology and a lexical resource
  This method implies usage of an existing structured lexical resource, like, for example, WordNet, and an alignment between its structure and the structure of the ontology.

Prevot *et al.* (2005) promote the third option as being the most complete of the listed approaches. The authors claim that this method should be followed in order to produce a system that is ontologically sound and linguistically motivated.

In information science, a useful distinction was made between foundational ontologies and domain ontologies. A foundational ontology models general concepts like *space*, *time*, *object*, *event*, etc., which are applicable across most of the domains. A domain-specific ontology models a specific domain, e.g., geography, and represents the particular meanings of terms as they apply to that domain. In the following, we will outline the main features of these two types of ontologies and review the existing ontologies, which are influential in the research on inference-based approaches to semantics.

### 3.2.1 *Foundational Ontologies*

As stated above, a foundational (or upper, or top-level ontology) ontology introduces general concepts that are the same across most knowledge domains. Every foundational ontology adopts a specific view on how the world should be conceptualized. Following Masolo *et al.* (2003), we list the typical ontological choices constituting meta-criteria for classifying ontologies.

- A *descriptive* ontology aims at capturing the ontological assumptions underlying natural language and cognition, while a *revisionary* ontology tries to capture the intrinsic nature of the world.
- A *multiplicative* ontology allows different entities to be co-localized in the same space-time, e.g., a vase is considered to be co-localized with the amount of clay, from which this vase has been made. A *reductionist* ontology rather claims that each space-time location contains at most one object. Thus, one can look at the same spatio-temporal entity from different points of view.
- A fundamental ontological choice concerns *the notion of change*. It implies a three-dimensional (3D) or a four-dimensional (4D) conceptualization of the world. In the 3D paradigm, physical objects are extended in space only, they are wholly present at each moment of their life, and are changing entities, in the sense that at different times they can instantiate different properties. In contrast, a 4D perspective implies that objects are also extended in time, only partially present at each moment, and are changing entities, in the sense that at different phases they can have different properties.
- Another ontological choice concerns the *actualism vs. possibilism* dichotomy. Actualism claims that only what is real exists. Possibilism admits possibilia (many different situations or worlds) as well. Possibilism implies that the representation language is required to express modalities, which typically means using a modal logic.

Ontological choices made by engineers of a foundational ontology force more specific ontologies, which will use this top-level ontology as a basis, to adopt the same principles of modeling the world. In this sense, a theoretically grounded upper ontology can imply a modeling methodology suggesting conceptual tools for choosing modeling options.

One of the main purposes of foundational ontologies is to support semantic interoperability between different more specific ontologies. Domain ontologies, which use the same upper ontology as a source of the basic categories employed to specify the meaning of the domain-specific elements, can be more easily merged automatically than ontologies having incompatible underlying conceptualizations. Thus, a foundational ontology should be universal enough so that every specific concept of a given domain could be suitably linked to a concept of the upper ontology.

#### 3.2.1.1 *Descriptive Ontology for Linguistic and Cognitive Engineering*

The Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE[29]; Masolo *et al.*, 2003) aims at capturing the upper ontological categories underlying natural language and human common sense. As its name suggests, DOLCE is descriptive, i.e. its purpose is not to capture the nature of the real-world entities, but rather to describe cognitive entities crucial for human perception and communication. According to the classification criteria above, DOLCE is multiplicative and follows possibilism.

DOLCE allows modeling 3D objects called *endurants* as well as 4D objects, *perdurants*. Roughly speaking, endurants correspond to physical objects; they are strictly identical at every time moment. Perdurants correspond to events; they happen in time. Perdurants can participate in endurants, for example, a person can participate in a discussion. The most general categories in DOLCE are *endurant*, *perdurant*, *quality*, and *abstract*. Qualities are described in DOLCE as “the basic entities we can perceive or measure: shapes, colors, sizes, sounds, smells, as well as weights, lengths” (Masolo *et al.*, 2003). The values of the qualities are referred to as *qualies* (e.g., *cold*, *red*, *50 kg*). Thus, a quality of a particular object (e.g., *the diameter of the moon*) is ontologically distinguished from its value (*3476 Km*). Time and space are also classified as qualities in DOLCE. Abstract entities in DOLCE are entities that do not have spatial nor temporal qualities and are not qualities themselves (e.g., facts, sets).

DOLCE includes not only a hierarchy of concepts, but also extensively axiomatized relations as well as spatial and temporal theories (e.g., axioms about parthood, participation, temporal-spatial coincidence). DOLCE axioms require a modal logic for their representation.

The structure of DOLCE is based on the OntoClean methodology (Guarino and Welty, 2004).[30] OntoClean represents an attempt to create criteria for evaluation of ontological taxonomies from a philosophical point of view applying techniques, which philosophers use to analyze, support, and criticize ontological descriptions of the world. OntoClean was successfully applied for cleaning up other ontologies (see, for example, Bateman *et al.*, 1995), and lexical-semantic resources such as WordNet (Oltramari *et al.*, 2002).

DOLCE is conceptually sound and explicit about its ontological choices. However, its taxonomy is not extensive and detailed enough for being directly employed in inference-

[29] *http://www.loa-cnr.it/DOLCE.html*
[30] *http://www.ontoclean.org/*

based NLP. In NLP research, DOLCE has been mainly used for interfacing domain-specific ontologies (see, for example Oberle *et al.*, 2007).

#### 3.2.1.2 *Suggested Upper Merged Ontology*

The Suggested Upper Merged Ontology (SUMO[31]) is an integrative database created "by merging publicly available ontological content into a single structure" (Niles and Pease, 2001). The top-level of SUMO is a merging of the Sowa's upper-level ontology (Sowa, 2000), the Russell and Norvig's upper-level ontology (Russell *et al.*, 1995), DOLCE (Masolo *et al.*, 2003), Allen's temporal axioms (Allen, 1984), Smith's ontology of boundaries (Smith, 1996), ontologies available on the Ontolingua server, and other resources (cf. Niles and Pease, 2001).

SUMO does not propose its own ontological principles but rather aims at benefiting from the existing ones. SUMO does not clearly adopt either a multiplicative or a reductionist approach. According to Oberle *et al.* (2007), the major part of the SUMO theories commit to the multiplicative view. Similarly, SUMO remains implicit concerning actualism/possibilism and 3D/4D choices. Oberle *et al.* (2007) classify SUMO as being descriptive because it adopts the commonsense distinction between objects and processes.

The most general categories in SUMO are *Physical* and *Abstract*. The former category describes everything that has a position in space and time, and the latter category includes everything else. SUMO further subdivides *Physical* into *Object* and *Process*. The taxonomy of SUMO is rich. In addition to the top-level ontology, the SUMO project includes a mid-level and several specific ontologies of such domains as communications, countries and regions, economy, finance, etc. SUMO contains 20 000 terms and 70 000 axioms when all domain ontologies are combined.

SUMO defines a hierarchy of classes, related rules and relationships, which are formulated in a version of SUO-KIF[32], language that has declarative semantics and goes beyond FOL. There exists also an OWL-Full[33] version of SUMO.[34]

According to Oberle *et al.* (2007), the conceptualization represented by SUMO is quite messy. Concepts can be instances at the same time, and some relations are modeled as concepts, e.g., there is a concept *BinaryRelation*. Nevertheless, the rich SUMO taxonomy

[31] *http://www.ontologyportal.org/*
[32] *http://suo.ieee.org/SUO/KIF/suo-kif.html*
[33] For more details about OWL representation language see Sec. 3.2.2.
[34] The SUMO group has organized a competition for reasoning systems providing the SUMO content as a text set (Pease *et al.*, 2008). The participating systems were required to verify the consistency of SUMO, and/or provide feedback to repair. The best performance was achieved by the Vampire reasoner (Riazanov and Voronkov, 2002).

makes it attractive for NLP researchers. In the last years, the SUMO project was quite active in the NLP area. There exists a mapping from WordNet to SUMO (Niles *et al.*, 2003). The result of this mapping is a list of synsets annotated with SUMO concepts. According to the classification proposed by Prevot *et al.* (2005), this work falls into the "populating" option, because no alignment of the structures of SUMO and WordNet has been performed.[35] The SUMO-WordNet mapping enabled the application of SUMO in the NLP context. For example, SUMO has been successfully employed in question answering (Harabagiu *et al.*, 2005; Suchanek, 2008).

#### 3.2.1.3 *OpenCyc*

OpenCyc[36] is a freely available ontology produced by the Cyc project, which started in 1984 with the aim of building a knowledge base including both scientific and commonsense knowledge (Guha and Lenat, 1990). The OpenCyc ontology does not follow any explicit philosophical principles or tested methodology; its focus is rather on coverage. The OpenCyc website reports that "The OpenCyc ontology now contains virtually all of of Cyc's hundreds of thousands of terms, along with millions of assertions relating the terms to each other, forming an upper ontology whose domain is all of human consensus reality". The 0.9 version of OpenCyc contains around 47 000 concepts and 300 000 facts about these concepts. OpenCyc is a clearly descriptive ontology. According to Prevot *et al.* (2005), it adopts the multiplicative approach, although this has not been followed in a systematic way. Similar to SUMO, OpenCyc does not explicitly choose between actualism vs. possibilism and 3D vs. 4D.

The knowledge in OpenCyc is represented using Second Order Predicate Logic. The project provides an inference system designed for reasoning with the OpenCyc content. There exist a mapping between OpenCyc and WordNet (Reed and Lenat, 2002). This mapping is obtained by adding the synonym relationship between OpenCyc concepts and WordNet synsets. Thus, this approach employs the populating method.

Currently, OpenCyc contains very little linguistic knowledge.[37] Therefore, up to now the resource has not attracted a lot of attention in the NLP community, although see, for example, Curtis *et al.* (2006).

[35] Prevot *et al.* (2005) criticize the resulting SUMO-WordNet mapping for the lack of methodology and evaluation.
[36] *http://www.opencyc.org/*
[37] Significantly greater NL functionality is provided by ResearchCyc, *http://www.cyc.com/cyc/cycrandd/areasofrandd_dir/nlu*, which is not freely available.

### 3.2.2 *Domain-specific Ontologies*

Development of domain-specific knowledge bases is motivated by many different AI tasks going far beyond NLP (cf. Guarino, 1998). Domain-specific ontologies contain concepts and relations belonging to a particular domain, objects that instantiate these concepts and relations as well as domain-specific theories. Usually a domain ontology consists of a so-called terminological box containing axioms that provide definitions for concepts and relations and an assertion box containing names of objects instantiating these concepts and relations. For example, in the domain of geography there might be a concept PACIFIC_ISLAND instantiated by the object *Tahiti*. The terminological box of this ontology might include the axiom $\forall x\ (\text{PACIFIC\_ISLAND}(x) \rightarrow \text{ISLAND}(x) \wedge \exists y\ (\text{LOCATEDIN}(x,y) \wedge \text{PACIFIC\_OCEAN}(y)))$, which means that a Pacific island is an island located in the Pacific ocean. The assertion box contains the assertion PACIFIC_ISLAND(*Tahiti*). Additionally, the ontology can include a theory consisting of more sophisticated axioms describing complex relationships between domain-specific concepts, relations, or objects, for example, $\forall x,y(\text{FRENCH\_POLYNESIA\_ISLAND}(x) \wedge \text{HAWAII\_ISLAND}(y) \rightarrow southern\_of(x,y))$, which means that if $x$ is an island located in French Polynesia and $y$ is a Hawaii island than $x$ is located to the south of $y$.

Extensive development of domain-specific ontologies was stimulated by the progress of Semantic Web technologies.[38] The Semantic Web (SW) project was originally focused on machine-understandable Web resources providing information, which can then be shared and processed by automated tools. Later on, the developed SW representation standards and corresponding reasoning tools also became popular in research areas that are not concerned with web documentation. A SW format called Web Ontology Language (OWL[39]; McGuinness and van Harmelen, 2004) was designed especially for the representation of ontological knowledge in a structured and formally well-understood way. OWL is logically based on Description Logics (DLs; Baader *et al.*, 2003), a family of model-theoretic knowledge representation languages that are especially attractive because of their respectively low computational complexity (see Sec. 4.4 for more details). Mostly exploited DLs belong to a decidable subclass of first-order logic. Because of the attractive combination of expressivity and low computational complexity of Description Logics, many DL-specific inference machines have been developed[40], which have enabled different AI applications to query OWL ontologies.

[38] *http://www.w3.org/2001/sw/*
[39] *http://www.w3.org/TR/owl-features/*
[40] A list of the existing reasoning tools can be found at *http://www.cs.man.ac.uk/~sattler/reasoners.html*.

Currently hundreds of domain-specific ontologies modeling different domains of applications are available on the web.[41] Domain-specific ontologies have been widely used in a variety of natural language applications playing a central role as a knowledge source for reasoning. A few examples of the kinds of NLP applications that employ reasoning with ontologies include the following:

- information retrieval (e.g., Andreasen and Nilsson, 2004; Buitelaar and Siegel, 2006)
- question answering (see Mollá and Vicedo, 2007, for an overview)
- dialog systems (e.g., Estival *et al.*, 2004)
- automatic summarization (e.g., Morales *et al.*, 2008)

## 3.3 Mixed Resources

Although the difference between lexical-semantic resources and ontologies seems to be clear (relations between words vs. relations between conceptual entities), there exist resources, which can only with difficulty be assigned to one or another category. These resources are ontologies, which have been generated automatically. Similarly to lexical-semantic resources, such ontologies operate with words rather than with artificially created labels. The difference is that they aim at providing conceptual knowledge for terms rather than grasping semantic relations between words. Automatically learned ontologies can be classified according to the learning method.

### 3.3.1 *Ontologies Learned from Text*

Ontology learning from text is a very active area of research, and many different systems performing this task have been developed, see Buitelaar and Cimiano (2008) for an overview. Two examples of such systems are OntoLearn (Velardi *et al.*, 2006) and Text2Onto (Cimiano and Völker, 2005). Both systems include similar processing steps, which are the following:

- Term extraction
  Simple and multi-word expressions relevant for a domain of interest are extracted from domain-related corpora, e.g., *integration strategy*.
- Extraction of relations between terms
  Relations between terms are extracted using patterns, e.g., *X is an Y*.

[41] Many of them can be accessed with a help of the Swoogle search engine, *http://swoogle.umbc.edu/*.

- Construction of definitions

  Extracted relations are combined into definitions represented in a machine-readable form, e.g., $\forall x(\textsc{Integration\_strategy}(x) \rightarrow \textsc{Strategy}(x) \wedge \exists y(\textsc{Integration}(y) \wedge purpose\_of(x,y)))$.

This approach is better suited for learning of domain-specific rather than general purpose ontologies. General terms are usually highly ambiguous, e.g., recall different meanings of *tragedy*; moreover, the same concept can be represented by different terms, e.g., *sick* and *ill*. In order to cope with these problems, automatic word sense disambiguation and detection of synonyms should be performed, which is far from trivial. In a closed domain, terminology is usually better defined, stable, and unambiguous. Therefore there is hope that definitions generated automatically using a domain-specific corpus will contain fewer mistakes than those extracted from general purpose corpora.

Thesauri are considered to be the best source for extraction of terminological knowledge, because they already contain explicit definitions for terms. Probably the most popular source of ontological knowledge is currently Wikipedia[42]. Today, its English version contains around 3 500 000 articles; the resource is growing daily.

Several approaches to automatic ontology learning employ first sentences of the Wikipedia articles. In such approaches, the title of the article corresponds to the concept label, while the first sentence is supposed to contain the concept definition. This sentence is parsed and the parse is converted into a machine-readable form (see, for example, Völker *et al.*, 2007). Although this method allows us to obtain more precise and structured definitions, it is also faces the problem of synonyms and ambiguity. Ontology definitions constructed in this way depend on concrete lexical and syntactic surface representations and contain words rather than of conceptual primitives.

### 3.3.2 *Ontologies Learned from Structured Sources: YAGO*

Another paradigm of ontology learning exploits structured machine-readable resources. For example, one of the largest of the existing freely available semantic databases wittily titled Yet Another Great Ontology (YAGO[43]; Suchanek *et al.*, 2007) utilizes Wikipedia's category pages. Category pages are lists of articles that belong to a specific category (e.g., Tahiti is in the category of islands). These lists give the YAGO project candidates for objects (e.g., *Tahiti*), candidates for concepts (e.g., ISLAND) and candidates for relations

[42] *http://www.wikipedia.org*
[43] *http://www.mpi-inf.mpg.de/yago-naga/yago/*

(e.g., *location*(*Tahiti*, *Pacific_ocean*)). In order to structure its content, YAGO links the concepts extracted from Wikipedia to the hierarchy of WordNet. YAGO currently includes more than 2 million entities and around 20 million facts about these entities. The resource mostly contains information about proper names (like persons, organizations, cities, etc.).

In contrast to most of the automatically learned resources, YAGO provides information of a high quality; it has a manually confirmed accuracy of 95%. In NLP, YAGO has been used, for example, for coreference resolution (Delmonte *et al.*, 2009; Bryl *et al.*, 2010).

### 3.3.3 *Ontologies Generated Using Community Efforts: ConceptNet*

In the last years, using community efforts for annotation, knowledge generation, or evaluation has become popular in the NLP area.[44] Concerning ontology generation, a good example of a resource created using community efforts is ConceptNet. ConceptNet[45] (Liu and Singh, 2004) is a machine-readable common sense knowledge resource automatically mined out of the Open Mind Common-sense corpus (OMSC[46]; Singh *et al.*, 2002). The idea of the OMSC project is that every person can contribute commonsense knowledge. With the help of the users of the Open Mind Common Sense web site, who were asked to enter sentences in a fill-in-the-blank fashion (e.g., "The effect of eating food is ..."; "A knife is used for ..."), the project has collected over 700 000 English sentences of commonsense. By applying NLP and pattern-based extraction rules to the OMCS sentences, 300 000 concepts and 1.6 million relations have been extracted to form ConceptNet's semantic network knowledge base.

ConceptNet nodes are natural language fragments, which are semi-structured according to preferred syntactic patterns, which fall into three classes: a) noun phrases denoting things, places, people, e.g., *life of party*, b) attributes, e.g., *very red*, and c) activity phrases, e.g., *get into accident*. Currently, ConceptNet contains 19 binary relations, which largely reflect the original choice of templates used on the OMCS web site, e.g., *PropertyOf*, *First-SubeventOf*, *LocationOf*.

The ConceptNet conceptual network is supplemented with a toolkit and API, which supports practical commonsense inferences about text, such as context finding, inference chaining, and conceptual analogy. The context finding function enables extraction of the contextual neighbors for a concept, for example, *take off clothes*, *go to sleep*, and *lie down*

[44] One of the most successful frameworks for employing joint human intelligence to perform tasks, which computers are unable to do, is the Amazon Mechanical Turk project, *https://www.mturk.com/mturk/*.
[45] *http://csc.media.mit.edu/conceptnet*
[46] *http://openmind.media.mit.edu/*

are neighbors of the *go to bed* term. Inference chaining feature supports finding paths in the semantic network, e.g., *buy food* $\rightarrow$ *have food* $\rightarrow$ *eat food* $\rightarrow$ *feel full* $\rightarrow$ *feel sleepy*. Getting conceptual analogy means finding concepts, which are structurally similar, e.g., *funeral* and *wedding* or *couch*, *sofa*, and *bed*.

Being a relatively recent project, ConceptNet has already attracted a lot of attention in the NLP community, because of the interesting combination of linguistic and conceptual knowledge. For example, it has been employed in question answering (Hsu *et al.*, 2008) and recognizing textual entailment (Arya *et al.*, 2010).

## 3.4 Concluding Remarks

In this chapter, the reader has seen a variety of sources for machine-readable world knowledge. Far from all of the existing resources have been covered; we presented only those, which are the most popular in contemporary NLP research with a focus on inference-based processing. We conclude the chapter with a brief summary of the main distinctive features of the listed types of resources and add a few remarks concerning integration of knowledge for the sake of natural language understanding.

*Lexemes vs. concepts*

Lexical-semantic resources focus on semantic relations between words (or word senses), while ontologies are intended to give definitions to conceptual entities. Thus, modeling primitives in lexical-semantic databases are lexemes (e.g., *island*), whereas ontologies introduce artificially created concept labels (e.g., FRENCH_POLYNESIA_ISLAND). Being built up with lexemes makes lexical-semantic resources more applicable in NLP, while ontologies require an additional lexical interface to be mapped to linguistic structures.

*Expressivity*

Lexical-semantic databases usually impose little structure: Semantic relations are for the most part two-place predicates relating two words, e.g., *hypernym*(*dog*, *animal*). SRL and paraphrasing resources (e.g., FrameNet, VerbOcean) additionally introduce relations between arguments of the predicate structures, e.g., $cause(kill(x,y),die(y))$. But there is still not much space in this representation for defining complex relationships like, for example, the fact that a Pacific island is an island located in the Pacific ocean. In contrast to lexical-semantic databases, ontologies are designed for supporting a detailed axiomatization. This difference is reflected in the representation languages required to formalize lexical-semantic and ontological knowledge. For representing lexical-semantic relations, a fragment of FOL is enough. These relations involve no logical connectors or quantification (all variables can be quantified universally).[47] In contrast, ontology representation languages may go beyond FOL or at least require an expressive fragment of FOL such as OWL.

*Fuzziness vs. consistency*

Lexical-semantic resources reflect ambiguity inherent in every natural language lexicon. A lexeme can refer to different concepts and a concept can be expressed by different lexemes. Semantic relations defined on lexemes (or word senses) as well as "definitions" assigned to conceptual items formed from lexemes (such as, for example, synsets) often have prototypical rather than obligatory character and may lead to logical contradictions, e.g., recall the *furniture–wheelchair* example in Sec. 3.1. The plausibility of the statements expressed by lexical-semantic relations can often be measured in probabilistic terms. This especially concerns resources learned automatically. Any inference system designed for handing lexical-semantic knowledge should be able to cope with fuzziness and ambiguity as well as to account for probabilistic information. In contrast, most of the ontologies are intended to be unambiguous and consistent. There is often a specific background theory underlying ontological modeling of a domain of interest. Correspondingly, most of the ontology representation formalisms are unable to handle contradictions, but rather support detection and repair of logical inconsistencies. Similarly, these formalisms are usually not designed for including probabilistic knowledge.

Thus, lexical-semantic resources accommodate knowledge, which lies both in the area of linguistic competence and common sense. This knowledge is fuzzy and probabilistic. On the contrary, ontologies are theory-based models of a domain of interest, which are extensively axiomatized and intended to be consistent. Although there might be an informational overlap between lexical-semantic resource and ontologies[48], these resources are in large part disjoint.

A natural question arises: Which resources are more appropriate for being used in natural language processing? Lexical-semantic databases, ontologies, or both? Lexical-semantic relations seem to be not enough for representing detailed world knowledge. On-

[47] See Sec. 5 for more details on representation of lexical-semantic knowledge.
[48] For example, such large resources like WordNet and OpenCyc have a significant overlap in the taxonomy.

tologies usually have only a weak relation to natural language.[49] It is not surprising that several researchers working on knowledge-intensive natural language processing came to the conclusion that both types of the knowledge resources are relevant for NLP and can be perfectly used in combination, see, for example, (Nirenburg and Raskin, 2004; Poesio, 2005; Prevot *et al.*, 2010).

Among lexical-semantic resources, two classes can be distinguished: hand-crafted electronic dictionaries and automatically generated databases. As stated in Sec. 3.1, hand-crafted dictionaries provide both extensive lexical coverage and accurate semantic labeling. Automatically generated resources introduce noise and contain relatively poor semantic labeling, but allow us to handle huge amounts of constantly changing knowledge available, for example, through the World Wide Web. Again, one might ask what type of lexical-semantic resources is preferable. At first sight, using automatically generated databases seems to be a more promising solution, since knowledge learned automatically is based on corpus data rather than on a subjective view of lexicographers and is relatively cheap to obtain and update. However, the NLP practice shows that current methods for automatic learning of semantic relations fail to produce knowledge of sufficient quality, therefore large manually developed resources like WordNet are still by far more popular in knowledge-intensive NLP. Possibly, the situation will change in the future, but developing NLP systems "here and now" we can only count on resources being available so far. Again, the best solution seems to be integration of data obtained by different methods. Manually developed lexical-semantic resources can be enriched by knowledge automatically learned from corpora and equipped with probabilistic information, see (Suchanek *et al.*, 2007; Cao *et al.*, 2008) for examples of successive integration approaches.

Ontologies fall into foundational and domain-specific categories. Foundational ontologies include those, which have only a top-level part, like, for example, DOLCE, and those, which axiomatize general knowledge on a large scale, like SUMO or OpenCyc. The main purpose of the upper level ontologies (e.g., DOLCE) is to provide a conceptualization skeleton and modeling principles, which can be adopted by more specific resources. Because of complexity and high level of abstraction, these ontologies are usually never used in practical NLP reasoning. In contrast, large foundational ontologies find their way in NLP if mapped to a lexical database, see Sec. 3.2 for examples. Domain-specific ontologies contain knowledge of a particular domain and represent a valuable source of world knowledge

[49]The interested reader is referred to the critical comparison of WordNet, Cyc, and EDR by the developers of these resources, G. Miller, D. Lenat, and T. Yokoi (Lenat *et al.*, 1995).

for domain-specific applications, which cannot be obtained from general purpose ontologies.

Our conclusion is that all types of resources discussed in this chapter are relevant for NLP, especially with respect to knowledge-intensive tasks. The only exception concerns upper level ontologies (e.g., DOLCE), which are not designed to be directly applied to natural language inferences, but rather provide principles and methodologies for organizing more specific knowledge. Thus, we are arguing in favor of integration of lexical-semantic resources (both built manually and generated automatically) with foundational and domain-specific ontologies. In Chap. 5, 6, and 7, we propose a framework designed for integration of these types of knowledge within a single reasoning pipeline.

It is interesting to see, what combinations of knowledge resources are actually employed in practice. The web portal *RTE Knowledge Resources*[50] gives an idea about which resources have been used by the RTE systems that have participated in the last Recognizing Textual Entailment Challenges. It is not surprising that WordNet is the absolute leader with 42 usages over past 3 years, while no other resource has been used more than 11 times. WordNet has been used in combination with a small task or domain-specific ontology (Bos and Markert, 2006; Tatu *et al.*, 2006), with the automatically generated DIRT resource (Iftene and Balahur-Dobrescu, 2007; Clark and Harrison, 2008), and with rules extracted automatically from VerbOcean and Wikipedia (Mehdad *et al.*, 2009).

[50] *http://www.aclweb.org/aclwiki/index.php?title=RTE_Knowledge_Resources*

# Chapter 4

# Reasoning for Natural Language Understanding

As already mentioned in the introductory chapter of this book, inference-based approaches to natural language semantics rely on a) a semantic parser outputting logical representations called *logical forms*[1], b) a knowledge base (KB) providing axioms for reasoning, c) an inference machine supporting reasoning with logical forms and KB axioms.

Figure 4.1 replicates the generalized inference-based NLP pipeline presented in the introductory chapter. A text fragment is input to a semantic parser, which outputs its logical form. The logical form and the knowledge base constitute input for the inference machine. An external NLP application generates queries for the inference machine (e.g., asks it to check whether a logical form follows from the KB) and uses its output to solve specific tasks. The three processing components of this pipeline (semantic parser, inference machine and final application) are usually independent from each other and can be replaced by alternative systems.

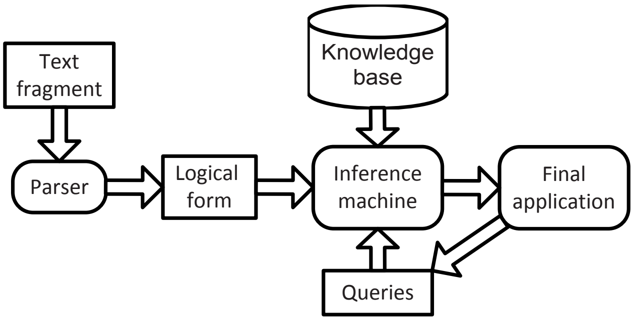


Fig. 4.1 Inference-based NLP pipeline

[1]Logical forms as understood in computational semantics are not to be confused with logical forms in the tradition of Generative Grammars, cf. Sec. 2.2.1.

In this chapter, we focus on semantic parsers and inference machines. In the study described in this book, we are not concerned with developing a new semantic parser, but use some of the existing systems, which are described in Sec. 4.1. Sections 4.2 and 4.3 are devoted to two different inference strategies, namely automatic deduction and abduction.

Reasoning with Description Logics is a special case of deductive reasoning. Since we elaborate on a decision algorithm for Description Logics (Sec. 6.2), this algorithm is described in detail in a separate section of this chapter (Sec. 4.4).

## 4.1 Semantic Parsers

In computational linguistics, *parsing* is a process of analyzing text in order to determine its structure and represent it in a machine-readable form. In most of the state-of-the-art parsing systems, text analysis is performed in a modular way. The text is sequentially processed by a tokenizer, a morphological tagger, a syntactic component, and, finally, a semantic component. Each component takes the output of the previous component as an input and outputs its own representation enriched by more linguistic information. Semantic parsing is usually the last stage of the text processing. It is supposed to output interpretations of text derived on the basis of an underlying semantic theory. In this section, we briefly introduce two semantic parsers.

### 4.1.1 *English Slot Grammar*

The English Slot Grammar (ESG) parser (McCord, 1990, 2010) converts English sentences into dependency trees that show both surface and deep structure. The deep structure is exhibited via a word sense predication for each node, with logical arguments.

*Slots* have two meanings in ESG. First, they represent syntactic roles, e.g., `subj` (subject), `obj` (object), `iobj` (indirect object). Second, they correspond to names of argument positions for predicates. For example, the ESG lexicon contains the following lexical slot frame for the verb *to give*: `give-vb'(e,x,y,z)`, which means “`e` is an eventuality where `x` gives `y` to `z`”. The logical form of the sentence *If Mary gives John a book then he reads it* in Fig. 4.2 contains the predicate `give-vb'(e3,x2,x6,x4)` such that its first argument `x2` points to *Mary*, the second argument `x6` points to *a book*, and `x4` points to *John*.[2]

[2]The ESG logical forms are given here in the style of Hobbs (1985b), see Sec. 5.1.1 for more details. They are conjunctions of propositions, which have generalized entity arguments that can be used for showing relationships among the predications.

```
if-in'(e1,e9,e3)
Mary-nn'(e2,x2)
give-vb'(e3,x2,x6,x4) Present-tense'(e11,e3)
John-nn'(e4,x4)
a'(e5,x6,e6)
book-nn'(e6,x6)
then-nn'(e7,x7) at-time'(e12,e9,e7)
he'(e8,x8)
read-vb'(e9,x8,x10) Present-tense'(e13,e9)
it'(e10,x10)
```

Fig. 4.2 The LF produced by *ESG* for the sentence *If Mary gives John a book, then he reads it.*

Generally, arguments in ESG are logical arguments. The first argument is always a node index. This corresponds to the event argument in Davidsonian representations (Davidson, 1967). The remaining arguments refer to the complement slot fillers of the corresponding predicate. They always come in the same order as defined in the lexical slot frame of this predicate. The logical form for the sentence *The book is given to John by Mary* is shown in Fig. 4.3. Note that the predicate `give-vb'(e4,e7,x2,e5)` has the argument `e7` *by Mary* as its logical subject (second argument) and has the node `x2` *the book* as its logical object (third argument) like in the representation in Fig. 4.2. Note that the logical connectors *if* and *then* are simply other predicates in the ESG output.

```
the'(e1,x2,e2)
book-nn'(e2,x2)
be_pass'(e3,x2,e4) Present-tense'(e9,e3)
give-vb'(e4,e7,x2,e5)
to-in'(e5,e4,x6)
John-nn'(e6,x6)
by-in'(e7,e4,x8)
Mary-nn'(e8,x8)
```

Fig. 4.3 The LF produced by *ESG* for the sentence *The book is given to John by Mary.*

### 4.1.2 *Boxer*

*Boxer*[3] (Bos, 2008) is a free semantic parser based on Combinatory Categorial Grammar (Steedman, 2001) and Discourse Representation Theory (Kamp and Reyle, 1993). It is able to construct Discourse Representation Structures (DRSs) for English texts, which can be then translated to first-order formulas.

Discourse Representation Theory (DRT) is a formal semantic theory designed to cope with intersentential anaphora and relations between the eventualities of different sentences. The theory covers a wide range of semantic phenomena such as anaphora, presupposition, tense and aspect, propositional attitudes, and plurals. DRT as introduced by Kamp and Reyle (1993) incorporates a Davidsonian event semantics (Davidson, 1967), where discourse referents can also stand for events and be referred to by anaphoric expressions or constrained by temporal relations. *Boxer* follows this approach using the inventory of roles proposed by VerbNet (Kipper *et al.*, 2000) in order to represent semantic roles.

An example of a DRS structure produced by *Boxer* for the sentence *If Mary gives John a book then he reads it* is shown in Fig. 4.4. The DRS represents the implicative structure of the sentence. The semantic roles *Agent*, *Theme*, *Recipient*, and *Topic* correspond to the semantic roles assigned to the verbs *to give* and *to read* in VerbNet.

```
  ___________________   ______________________________________
 |x0 x1 x2           | |                                      |
 |...................| |......................................|
(|named(x0,mary,per) |+|  ________________    ______________  |)
 |named(x1,john,per) | | |x3 e4           |  |e5            | |
 |neuter(x2)         | | |................|  |..............| |
 |___________________| | |book(x3)        |>|read(e5)       | |
                       | |give(e4)        |  |Agent(e5,x1)  | |
                       | |Agent(e4,x0)    |  |Theme(e5,x2)  | |
                       | |Recipient(e4,x1)|  |then(e5)      | |
                       | |Theme(e4,x3)    |  |______________| |
                       | |________________|                   |
                       |______________________________________|
```

Fig. 4.4 The DRS produced by *Boxer* for the sentence *If Mary gives John a book, then he reads it.*

The translation of the DRS structures into first-order logic syntax using the translation function $\cdot^{d2f}$ is defined as follows (cf. Blackburn *et al.*, 2001):

[3] *http://svn.ask.it.usyd.edu.au/trac/candc/wiki/boxer*

$$\left(\begin{array}{|c|}\hline x_1,\ldots,x_n \\ \hline \gamma_1 \\ \vdots \\ \gamma_m \\ \hline \end{array}\right)^{d2f} = \exists x_1,\ldots,x_n((\gamma_1)^{d2f} \wedge \ldots \wedge (\gamma_m)^{d2f})$$

$$R(x_1,\ldots,x_n)^{d2f} = R(x_1,\ldots,x_n)$$
$$(\tau_1 = \tau_2)^{d2f} = (\tau_1 = \tau_2)$$
$$(\neg B)^{d2f} = \neg(B)^{d2f}$$
$$(B_1 \vee B_2)^{d2f} = (B_1)^{d2f} \vee (B_2)^{d2f}$$

$$\left(\begin{array}{|c|}\hline x_1,\ldots,x_n \\ \hline \gamma_1 \\ \vdots \\ \gamma_m \\ \hline \end{array} \Rightarrow B\right)^{d2f} = \forall x_1,\ldots,x_n(((\gamma_1)^{d2f} \wedge \ldots \wedge (\gamma_m)^{d2f}) \rightarrow (B)^{d2f})$$

Given the translation rules above, the DRS in Fig. 4.4 will be translated into the following FOL representation:

$$\exists x0,x1,x2(named(x0,mary,per) \wedge named(x1,john,per) \wedge neuter(x2) \wedge$$
$$\forall x3,e4((give(e4) \wedge book(x3) \wedge Agent(e4,x0) \wedge Theme(e4,x3) \wedge Recipient(e4,x1)) \rightarrow$$
$$\exists e5(read(e5) \wedge Agent(e5,x1) \wedge Theme(e5,x2) \wedge then(e5))))$$

As the example in Fig. 4.4 shows, *Boxer* performs anaphora resolution and named entity recognition using such labels as `per`, `loc`, etc. In addition, the system allows us to map lexemes into WordNet synsets and VerbNet classes.[4] As reported by Bos (2008), *Boxer* reaches more than 95% coverage on newspaper texts.

## 4.2 Deduction for Natural Language Understanding

Deductive reasoning consists in showing that some statement (conclusion) is a logical consequence of a set of statements (premises). The conclusion is called *logically valid* if it follows from the premises. A deduction calculus consists of a set of logical axioms and a collection of deduction rules for deriving new formulas given previously derived formulas.

[4]No word sense disambiguation is performed; the most frequent word sense is assigned to each lexeme.

For example, given the axiom $\forall x(p(x) \rightarrow q(x))$ and an individual $A$ such that $p(A)$, one concludes $q(A)$.

In the field of automatic deduction (or theorem proving), most attention is paid to first-order logic (FOL), because many knowledge representation formalisms can be reasonably reduced to FOL. First-order logic is in general undecidable, i.e. not for every FOL formula it is possible to decide algorithmically whether it is valid or invalid.

The first major breakthrough in automatic reasoning was made by Robinson (1965) who has developed a resolution algorithm for FOL. The resolution algorithm works with formulas converted into conjunctive normal form (CNF).[5] All quantification becomes implicit in this form. Universal quantifiers are omitted and existentially quantified variables are replaced by *Skolem functions*.[6] The algorithm is defined as follows:

(1) find two clauses (conjuncts in CNF) containing the same predicate, where it is negated in one clause but not in the other;
(2) perform a unification on the two propositions;
(3) if any unbound[7] variables, which were bound in the unified predicates also occur in other predicates, replace them with their bound values;
(4) discard the unified predicates, and combine the remaining ones from the two clauses into a new clause, also joined by the $\wedge$ operator.

For example, the axiom $\forall x\ (p(x) \rightarrow q(x))$ is converted into $\neg p(x) \vee q(x)$. Given $p(A)$ as another premise, reasoning works as follows. Rule (1) above is applied and we find that the predicate $p$ occurs in negated form ($\neg p(x)$) in the first clause and in non-negated form ($p(A)$) in the second clause, $x$ is an unbound variable, and $A$ is a bound value (constant). Rule (2) forces the substitution $x \mapsto A$. According to rule (3), this substitution is applied to the remaining predicate $q(x)$ producing $q(A)$. Rule (4) discards the unified predicates and produces the conclusion $q(A)$.

The idea of applying automated deduction to natural language understanding originated in the context of question answering (Black, 1964; Green and Raphael, 1968). Later de-

[5]A statement is in conjunctive normal form if it is a conjunction consisting of one or more conjuncts, each of which is a disjunction of one or more (negated) atomic predicates.

[6]Skolemization is performed as follows. If an existentially quantified variable $x$ is not inside the scope of a universal quantifier then it can be replaced by creating new constants. For example, $\exists x\ P(x)$ can be changed to $P(c)$, where $c$ is a new constant. Otherwise, $x$ is replaced by $f(y_1, \ldots, y_n)$, where $f$ is a new function and $y_1, \ldots, y_n$ are universally quantified variables whose quantifiers precede that of $x$. For example, $\forall y\ \exists x\ P(x,y)$ can be changed to $\forall y\ P(f(y),y)$.

[7]An unbound (or free) variable is a placeholder, which can be later replaced by some specified value, e.g., constant. In an expression, it is a variable whose value must be known in order for the whole expression to be evaluated.

ductive reasoning was successfully employed for story understanding (Winograd, 1972; Charniak, 1972).

As summarized by Gardent and Webber (2001), there are two main current approaches to employing automated reasoning in natural language understanding. The first approach is to filter out unwanted text interpretations, for example, see Bos (2009). The main criterion used to identify and reject unwanted readings is consistency. A set of formulas $\Phi$ in first-order logic is *consistent* if and only if there is no formula $\phi$ such that $\Phi$ entails $\phi$ and $\Phi$ entails $\neg\phi$.

For example, consider sentences 4.1(a). There are two possible readings of these sentences corresponding to the two possible antecedences of the pronoun *it*: 4.1(b) and 4.1(c). In the first reading the dog is hungry and in the second reading the bone is hungry. Suppose the knowledge base contains axioms 4.1(d).

**Example 4.1.**

(a) *The dog ate the bone. It was hungry.*

(b) $\exists d,b,e(dog(d) \wedge eat(e,d,b) \wedge hungry(d))$

(c) $\exists d,b,e(dog(d) \wedge eat(e,d,b) \wedge hungry(b))$

(d)
$\forall x(hungry(x) \rightarrow living\_being(x))$ (*Only living beings can be hungry.*)
$\forall d(dog(d) \rightarrow living\_being(d))$ (*Dogs are living beings.*)
$\forall b(bone(b) \rightarrow \neg living\_being(b))$ (*Bones are not living beings.*)

If the resolution algorithm will be applied to this knowledge base and to each of the two readings above, then reading 4.1(c) will imply inconsistency and will therefore be rejected.

Another approach to integrating automated deduction into NLU implies that a more specific representation is constructed in the course of proving the underspecified one (see Bos, 2003; Cimiano, 2003). This approach is based on a model builder – a program that takes a set of logical formulas $\Phi$ and tries to build a model that satisfies $\Phi$.[8] This approach has the advantage of providing a consistency check "for free", since the only models that will be built are both logically consistent and consistent with world knowledge.[9]

Different criteria can be suggested to choose between alternative models. The most popular criterion is *minimality* (for a detailed consideration see Gardent and Webber, 2001;

[8] An *interpretation* is a mathematical structure that describes how the symbols of a logical theory are interpreted. An interpretation for a FOL language $\mathscr{L}$ is a pair $(\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$, where with $\Delta^{\mathscr{I}}$ a non-empty set of entities (the domain of individuals) and $\cdot^{\mathscr{I}}$ an interpretation function mapping relation symbols in $\mathscr{L}$ to relations of appropriate arity in $\Delta^{\mathscr{I}}$ and constant symbols in $\mathscr{L}$ to elements of $\Delta^{\mathscr{I}}$. An interpretation that satisfies a set of logical formulas $\Phi$ is called a *model* of $\Phi$.

[9] As it has been noticed by several researchers, model building is similar to the use of abduction for NLU as introduced in the next section (Baumgartner and Kühn, 2000; Gardent and Konrad, 2000).

Bos, 2003; Cimiano, 2003). The concept of minimality can be applied to (a) the number of entities in the domain of individuals, and (b) the number of positive predicate extensions (for predicates with an arity higher than 0).

For example, given sentences 4.2(a) with logical form 4.2(b) and knowledge base 4.2(c), $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$ are three possible models of 4.2(b).

**Example 4.2.**

(a) *John saw the house. The door was open.*

(b) $\exists x,e,y,z(John(x) \wedge see(e,x,y) \wedge house(y) \wedge door(z) \wedge open(z))$

(c) $\forall h(house(h) \rightarrow \exists d(door(d) \wedge part\_of(h,d)))$

$\mathcal{M}_1 = \{John(j), see(a,j,h), house(h), door(d_1), part\_of(h,d_1), door(d_2), open(d_2)\}$

$\mathcal{M}_2 = \{John(j), see(a,j,h), house(h), door(d), part\_of(h,d), open(d)\}$

$\mathcal{M}_3 = \{John(j), see(a,j,j), house(j), door(j), part\_of(h,j), open(j)\}$

$\mathcal{M}_2$ is preferred over $\mathcal{M}_1$, because $\mathcal{M}_2$ contains less individuals in the domain. $\mathcal{M}_3$ is even smaller than $\mathcal{M}_2$, however, this model is not intuitive, because it assigns the same individual $j$ to *John*, *house*, and *door*. In order to block it, one has to add additional axioms to the knowledge base stating that *John*, *house*, and *door* are disjoint. Stating constraints, which block unification of variables is far from straightforward; for problematic cases see, for example, (Gardent and Webber, 2001).

Blackburn and Bos (2005) show that state-of-the-art automated deduction can be of practical use in checking consistency for at least some classes of linguistic problems if theorem provers and model builders are used in parallel. In order to check the consistency of $\phi$, one simultaneously asks the theorem prover to prove $\neg\phi$ and the model builder to construct a model that satisfies $\phi$. If the theorem prover succeeds to prove inconsistency of $\neg\phi$ then $\phi$ is valid. If the model builder succeeds to build a model of $\phi$, then $\phi$ is satisfiable and therefore consistent.

Classical deduction cannot accommodate uncertain knowledge. For example, the axiom $\forall x(bird(x) \rightarrow \exists e(fly(e,x)))$ is not necessary (though typically) true, because there are birds that never fly. However, in a classical deductive framework, representing defeasible knowledge is impossible. In order to cope with this problem, Richardson and Domingos (2006) propose to use Markov Logic Networks for performing first-order inference in a probabilistic way. In this framework, a weight is assigned to every first-order axiom. A logical formula is taken as a Markov network so that the vertices of the network graph are atomic formulas, and the edges are the logical connectives used to construct the formula. It allows us to soften first-order logic by making situations in which not all constraints are satisfied less likely but not impossible. For an application of this approach to natural language understanding see, for example, Garrette *et al.* (2011) .

Automated reasoning is an active research area. Some state-of-the-art theorem provers are described by Robinson and Voronkov (2001).[10] In the experiments described in this book, the *Nutcracker* system[11] was used (Bos and Markert, 2006). *Nutcracker* is a system for recognizing textual entailment, developed by Johan Bos, see Sec. 8.2 for more details. It is based on the *Boxer* system (Sec. 4.1.2) and includes a free first-order, resolution based theorem-prover *BLIKSEM*[12], and two model builders *MACE* and *Paradox*. *MACE*[13] (Mccune, 2003) uses the Davis-Putnam-Loveland-Logeman decision procedure to construct models (Davis *et al.*, 1962). *Paradox* (Claessen and Sörensson, 2003) is using a similar technique but with further optimizations.

## 4.3 Abduction for Natural Language Understanding

In logics, abduction is inference to the best explanation. Given the axiom $\forall x(p(x) \rightarrow q(x))$ and an individual $A$ such that $q(A)$, one concludes $p(A)$. In this framework, $q(A)$ is an observation and $\forall x(p(x) \rightarrow q(x))$ is a general principle, which can be used to explain the occurrence of $q(A)$ so that $p(A)$ is assumed to be the underlying explanation of $q(A)$. Suppose you know the rule *If it rains then the grass is wet*. If the grass is wet indeed, one can conclude that it might have been raining as a possible explanation of why the grass is wet. Of course, this inference is not valid, because there may be several alternative explanations for $q(A)$, e.g., somebody watered the grass.

Two early approaches to using abductive reasoning in natural language understanding were developed by Norvig (1983) and Wilensky (1983). In these approaches, an operation of *concretion* is proposed, which is "the process of inferring a more specific interpretation of an utterance that is justified by language alone" (Wilensky *et al.*, 1988). For example, the result of concretion of the sentence *John gave Mary a kiss* is a more specific interpretation *John kissed Mary*. The concretion mechanism works as follows: "If all applicable conditions are met, the concept becomes an instance of the subcategory" (Wilensky *et al.*, 1988).

[10]Many theorem proving software systems are listed at *http://en.wikipedia.org/wiki/Category:Theorem_proving_software_systems*.
[11]*http://svn.ask.it.usyd.edu.au/trac/candc/wiki/nutcracker*
[12]*http://www.ii.uni.wroc.pl/~nivelle/software/bliksem/*
[13]*http://www.cs.unm.edu/~mccune/prover9/*

Norvig (1987) implemented this process as *marker passing* in a semantic net. Markers start out with a given amount of marker energy, and are passed through the network, losing energy with each pass, and stopping when the energy value is equal to zero. If two or more markers are passed to the same node then there is a marker collision. For each collision, the paths of the corresponding markers in the semantic net are inspected, and if they are of the right shape then an inference is suggested.

In later work, weighted, or cost-based, abduction is employed as a mechanism to choose between alternative abductive explanations (Charniak and Goldman, 1989; Stickel, 1990; Hobbs *et al.*, 1993). In this framework, any assumed formula is assigned a non-negative real number cost. The best explanation is then the proof with the minimum cost. In the following, we describe this approach in detail according to how it is presented by Hobbs *et al.* (1993).

In the framework of weighted abduction, input formulas are flat conjunctions of $n$-ary predications, such that a real number cost is attached to every predication as shown below:

$$q_1(x_1,\ldots,x_k) : c_1 \wedge \ldots \wedge q_n(y_1,\ldots,y_l) : c_n, \tag{4.1}$$

where $q_j$ ($1 \leq j \leq n$) are predicate names, $x_r$ and $y_s$ ($1 \leq r \leq k$, $1 \leq s \leq l$) are variables existentially quantified with the widest possible scope, and $c_j$ are non-negative real number costs. Axioms in the knowledge base have the following form[14]:

$$P_1^{w_1} \wedge \ldots \wedge P_m^{w_m} \rightarrow Q_1 \wedge \ldots \wedge Q_n, \tag{4.2}$$

where $P_i$ ($1 \leq i \leq m$) and $Q_j$ ($1 \leq j \leq n$) are propositions of the form $predicate\_name(x_1,\ldots,x_k)$ and $w_i$ are non-negative real numbers. The axiom above states that the conjunction of $P_i$ implies the conjunction of $Q_j$. Therefore, given the conjunction of $Q_j$ occurring in the input, such that the cost $c_j$ is attached to every $Q_j$, the conjunction of $P_i$ can be assumed. It also states that if the total cost of all $Q_j$ is $c = \sum_{j=1}^{n} c_j$, then the cost of assuming each $P_i$ is $f(w_i, c)$, where $f$ is some arbitrary function. After an axiom was applied to a proposition $q$, the cost of $q$ will be set to 0, because its cost is now carried by the newly introduced assumptions. An axiom can be applied only to a conjunction of propositions containing at least one proposition with a non-zero cost.

Given input Eq. 4.1 and axiom $p(y_1,z)^w \rightarrow q_n(y_1,\ldots,y_l)$, the result of the application of the axiom to the formula is as follows:

[14] In this framework, all variables occurring on the left-hand side of an axiom are universally quantified, while variables occurring on the right-hand side only are existentially quantified. Hereinafter, we do not use quantification in the abductive axioms.

$$q_1(x_1,\ldots,x_k):c_1 \wedge \ldots \wedge q_n(y_1,\ldots,y_l):0 \wedge p(y_1,z):f(w,c_n), \tag{4.3}$$

where $f(w,c_n)$ is the cost of the newly introduced proposition $p(y_1,z)$ and the first argument of $p$ is equal to the first argument of $q_n$. The cost of $q_n(y_1,\ldots,y_l)$ is now equal to 0.

Weighted abduction supports factoring. If two propositions in a formula have the same name and arity, they can be unified so that the resulting proposition is given the smaller of the costs of the input propositions. For example, given the formula

$$u(x_1):c_u \wedge \ldots \wedge q(x_n):20 \wedge \ldots \wedge q(x_m):10 \wedge \ldots \wedge v(x_k):c_v, \tag{4.4}$$

where $q(x_n)$ costs 20 and $q(x_m)$ costs 10, the procedure assumes $x_n$ and $x_m$ to be identical and outputs the following formula:

$$u(x_1):c_u \wedge \ldots \wedge q(x):10 \wedge \ldots \wedge v(x_k):c_v, \tag{4.5}$$

where $q(x)$ costs 10.[15] This feature allows us to exploit implicit redundancy made explicit by a proof. The idea behind it is that if an assumption has already been made then there is no need to make it again.

Since the abductive approach is based on backward-chaining rather than forward-chaining (given $P \rightarrow Q$, $P$ is inferred from $Q$), it might seem that it is impossible to use superset information in this framework, e.g., that *dog* implies *animal*. In order to cope with this problem, superset axioms should be converted into biconditional. Thus, axioms of the form

$species \rightarrow genus$

can be converted into axioms of the form

$genus \wedge\ differentiae \leftrightarrow species$,

where differentiae often cannot be proven or even specified. In the abductive framework this is not a problem, because differentiae can be simply assumed instead of being proven. One can specify axioms like

$animal(x)^{0.2} \wedge etc(x)^{0.9} \rightarrow dog(x)$,

where the *etc* ("et cetera") proposition stands for all properties of an animal, which are required to be a dog.

[15]Note that setting variables to be identical can influence other predicates, which have these variables as their arguments.

Hobbs *et al.* (1993) describe how weighted abduction can be applied to the discourse processing problem viewing the process of interpreting sentences in discourse as the process of providing the best explanation of why the sentence would be true. In this framework, interpreting a sentence means proving its logical form, merging redundancies where possible, and making assumptions where necessary.

The example below shows how abductive reasoning can be applied to lexical disambiguation. Suppose we want to construct an interpretation of the sentence *John composed a sonata* translated by a semantic parser into the following logical form:

$John(x_1) \wedge compose(e, x_1, x_2) \wedge sonata(x_2)$

The lexeme *compose* is ambiguous between the "put together" reading instantiated, for example, in the sentence *The party composed a committee*, and the "create art" meaning instantiated in the sentence above.

Suppose our knowledge base contains the following axioms:[16]

(1) $put\_together(e, x_1, x_2)^{1.2} \rightarrow compose(e, x_1, x_2)$
(2) $create\_art(e, x_1, x_2)^{0.6} \wedge work\_of\_art(x_2)^{0.6} \rightarrow compose(e, x_1, x_2)$
(3) $sonata(x)^{1.5} \rightarrow work\_of\_art(x)$

Axioms (1) and (2) correspond to the two readings of *compose*. Axiom (3) states that sonata is a work of art.

The logical form and the axioms are input into an abductive reasoner, which is supposed to construct the best interpretation of the logical form by applying axioms to it. In order to start reasoning, we need a) to define the costs of the input propositions, b) to define the cost function $f$. For the sake of example, let the costs of the input propositions *John*, *compose*, and *sonata* be equal to 20 and $f(w, c) = w \cdot c$.

Two interpretations can be constructed for the logical form above. The first one is the result of the application of axiom (1).[17] Note that the costs of the backchained proposition *compose* is set to 0, because its cost is now carried by the newly introduced assumption *put_together*. The total cost of the first interpretation **I1** is equal to 64.

**I1**: $John(x_1) : 20 \wedge compose(e, x_1, x_2) : 0 \wedge sonata(x_2) : 0 \wedge$
$put\_together(e, x_1, x_2) : 24$

[16] The axiom weights in the given example are arbitrary.
[17] The numbers after every proposition correspond to the costs of these propositions.

The second interpretation is constructed in several steps. First, axiom (2) is applied, so that *compose* is backchained on to *create_art* and *work_of_art* with the costs equal to 12. Then, axiom (3) is applied to *work_of_art*.

$$\mathbf{I2}_1: \quad John(x_1):20 \wedge compose(e,x_1,x_2):0 \wedge sonata(x_2):20 \wedge$$
$$create\_art(e,x_1,x_2):12 \wedge work\_of\_art(x_2):0 \wedge sonata(x_2):18$$

The total cost of $\mathbf{I2}_1$ is equal to 70. This interpretation is redundant, because it contains the proposition *sonata* twice. The procedure will merge propositions with the same predicate name, setting the corresponding arguments of these propositions to be equal and assigning the minimum of the costs to the result of merging. The final form of the second interpretation $\mathbf{I2}_2$ with the cost of 60 is as follows. The "create art" meaning of *compose* was chosen because it reveals implicit redundancy in the sentence.

$$\mathbf{I2}_2: \quad John(x_1):20 \wedge compose(e,x_1,x_2):0 \wedge sonata(x_2):18 \wedge$$
$$create\_art(e,x_1,x_2):12 \wedge work\_of\_art(x_2):0$$

Thus, on each reasoning step the abductive reasoning procedure:

(1) merges propositions with the same predicate name and arity, assigning the lowest cost to the result of merging,
(2) applies an axiom to a conjunction of propositions such that one of these propositions has a non-zero cost.

Hobbs *et al.* (1993) construct examples showing that the described procedure provides solutions to a whole range of natural language pragmatics problems, such as resolving reference, ambiguity and metonymy, discovering implicit relations in nouns compounds and prepositional phrases, or making discourse structure explicit. This approach has the expressivity of logical inference but also supports probabilistic, fuzzy, or defeasible reasoning and includes measures of the "goodness" of abductive proofs and hence of interpretations of texts.

To sum up, abductive reasoning is promising for discourse processing, because this approach:

- exploits redundancy in natural language texts;
- provides solution to several natural language pragmatics problems as a by-product;
- allows assumptions, i.e. reasoning with incomplete knowledge;
- provides a solution to the problems of nonmonotonicity and vagueness;

- provides a measure of the "goodness" of obtained interpretations, which can be also seen as a measure of text coherence;
- solves the problem of where to stop drawing inferences, which could easily be unlimited in number; an inference is appropriate if it is part of the lowest-cost proof of the logical form.

One drawback of this approach is that its practical application crucially depends on heuristics. Applying weighted abduction implies defining a) costs of the input propositions, b) the cost function $f$, and, most importantly, c) axiom weights. It is easy to see that all these three factors can crucially influence the reasoning process; this especially concerns axiom weights. Unfortunately, no theoretically grounded solution has been provided for these problems yet. Since in this book we are concerned with practical application of abductive reasoning to discourse processing, we focus on the issue of defining axiom weights in detail and propose a frequency-based method of calculating weights (Chap. 5).

The approach described in this section was first implemented as *TACITUS* system (Hobbs *et al.*, 1993), which was based on Stickel's Prolog Technology Theorem Prover (*PTTP*, Stickel, 1988). It was later re-implemented as *Mini-TACITUS* (Mulkar *et al.*, 2007) – a simple backchaining theorem-prover intended to be a more transparent version of the original system. We extended *Mini-TACITUS* in order to make it able to reason with a large knowledge base as described in Sec. 7.

## 4.4 Reasoning with Description Logics

As described in Sec. 3.2, many of the modern knowledge-based NLU systems use knowledge bases represented in the form of ontologies; this especially concerns domain-specific applications. Nowadays, the main format for representing ontological knowledge is called Web Ontology Language (OWL[18]; McGuinness and van Harmelen, 2004). OWL is logically founded on Description Logics (DLs), a family of model-theoretic knowledge representation languages that represent a decidable subset of first-order logic (Baader *et al.*, 2003). In this section, we briefly overview notation and terminology of Description Logics as well as a corresponding reasoning algorithm. For comprehensive background reading, the reader is referred to Baader *et al.* (2003).

In Description Logics, the main expressive means are called *concept descriptions*. Given a set $N_C$ of *concept names* and a set $N_R$ of *role names*, concept descriptions are

[18] *www.w3.org/TR/owl-features/*

Table 4.1 Syntax and semantics of $\mathscr{ALCN}$ DL.

| **Constructor** | **Syntax** | **Semantics** |
|---|---|---|
| negation | $\neg C$ | $\Delta^{\mathscr{I}} \setminus C^{\mathscr{I}}$ |
| conjunction | $C_1 \sqcap C_2$ | $C_1^{\mathscr{I}} \cap C_2^{\mathscr{I}}$ |
| disjunction | $C_1 \sqcup C_2$ | $C_1^{\mathscr{I}} \cup C_2^{\mathscr{I}}$ |
| existential restriction | $\exists R.C$ | $\{x \in \Delta^{\mathscr{I}} \vert \exists y : (x,y) \in R^{\mathscr{I}} \wedge y \in C^{\mathscr{I}}\}$ |
| value restriction | $\forall R.C$ | $\{x \in \Delta^{\mathscr{I}} \vert \forall y : (x,y) \in R^{\mathscr{I}} \rightarrow y \in C^{\mathscr{I}}\}$ |
| at-least restriction | $\geq nR.C$ | $\{x \in \Delta^{\mathscr{I}} \vert\, \vert\{y \in \Delta^{\mathscr{I}} \vert (x,y) \in R^{\mathscr{I}} \wedge y \in C^{\mathscr{I}}\}\vert \geq n\}$ |
| at-most restriction | $\leq nR.C$ | $\{x \in \Delta^{\mathscr{I}} \vert\, \vert\{y \in \Delta^{\mathscr{I}} \vert (x,y) \in R^{\mathscr{I}} \wedge y \in C^{\mathscr{I}}\}\vert \leq n\}$ |
| top concept | $\top$ | $\Delta^{\mathscr{I}}$ |
| bottom concept | $\perp$ | $\varnothing$ |

defined with the help of *concept constructors* determining the expressive power of the DL in question. The $\mathscr{ALCN}$ DL logic relevant for the ontologies considered in this book, is defined in Table 4.1, where $C \in N_C$, $R \in N_R$, $n \in \mathbb{N}$.

The semantics of concept descriptions is defined in the usual way in terms of an *interpretation* $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$, where $\Delta^{\mathscr{I}}$ is a non-empty set of individuals and the function $\cdot^{\mathscr{I}}$ maps every concept name $A$ to $A^{\mathscr{I}} \subseteq \Delta^{\mathscr{I}}$ and every role name $R$ to $R^{\mathscr{I}} \subseteq \Delta^{\mathscr{I}} \times \Delta^{\mathscr{I}}$.

A DL ontology consists of a *terminological box* (TBox) and *assertion box* (ABox). A TBox $\mathscr{T}$ is a set of axioms of the form $C \sqsubseteq D$ or $C \equiv D$, where $C$ and $D$ are concept descriptions. An interpretation $\mathscr{I}$ is a *model* of $C \sqsubseteq D$ if $C^{\mathscr{I}} \subseteq D^{\mathscr{I}}$; $\mathscr{I}$ is a model of $C \equiv D$ if $C^{\mathscr{I}} = D^{\mathscr{I}}$. $\mathscr{I}$ is a model of a TBox $\mathscr{T}$ if it is a model of every axiom in $\mathscr{T}$.

Given a set of $N_I$ individual names, an ABox $\mathscr{A}$ is a finite set of assertions of the form $C(a)$ (concept assertions) or $R(a,b)$ (role assertions), where $C$ is a concept description, $R$ is a role name, and $a,b \in N_I$. An interpretation $\mathscr{I}$ is a model of $C(a)$ if $a^{\mathscr{I}} \in C^{\mathscr{I}}$; $\mathscr{I}$ is a model of $R(a,b)$ if $\langle a^{\mathscr{I}}, b^{\mathscr{I}} \rangle \in R^{\mathscr{I}}$. $\mathscr{I}$ is a model of $\mathscr{A}$ if it is a model of every axiom in $\mathscr{A}$.

An interpretation $\mathscr{I}$ is a model of a knowledge base $\mathscr{K} = (\mathscr{T}, \mathscr{A})$ if $\mathscr{I}$ is a model of $\mathscr{T}$ and $\mathscr{I}$ is a model of $\mathscr{A}$.

The most important inference services provided by DL ontologies concern computing satisfiability, subsumption, and consistency. A knowledge base $\mathscr{K} = (\mathscr{T}, \mathscr{A})$ is *consistent* if it has a model. A concept description $C$ is *satisfiable* towards $\mathscr{T}$ if there exists a model

Table 4.2 Tableau expansion rules for $\mathscr{ALCN}$ satisfiability.

| | |
|---|---|
| Rule $\sqcap$ | if $\mathscr{A}$ contains $(C_1 \sqcap C_2)(x)$, but not both $C_1(x)$ and $C_2(x)$<br>then $\mathscr{A}' := \mathscr{A} \cup \{C_1(x), C_2(x)\}$ |
| Rule $\sqcup$ | if $\mathscr{A}$ contains $(C_1 \sqcup C_2)(x)$, but neither $C_1(x)$ nor $C_2(x)$<br>then $\mathscr{A}' := \mathscr{A} \cup \{C_1(x)\}, \mathscr{A}'' := \mathscr{A} \cup \{C_2(x)\}$ |
| Rule $\exists$ | if $\mathscr{A}$ contains $(\exists R.C)(x)$, but there is no individual $y$ such that $C(y)$ and $R(x,y)$ are in $\mathscr{A}$<br>then $\mathscr{A}' := \mathscr{A} \cup \{C(y), R(x,y)\}$, where $y$ is a new individual not occurring in $\mathscr{A}$ |
| Rule $\geq$ | if $\mathscr{A}$ contains $(\geq nR)(x)$, and there are no individuals $y_1, \ldots, y_n$<br>such that $R(x,y_i) \in \mathscr{A}$ and $y_i \neq y_j$ $(i, j \in \{1, \ldots, n\}$ and $j \neq i)$<br>then $\mathscr{A}' := \mathscr{A} \cup \{R(x,y_i \mid 1 \leq i \leq n)\}$,<br>where $y_1, \ldots, y_n$ are distinct individuals no occurring in $\mathscr{A}$ |
| Rule $\leq$ | if $\mathscr{A}$ contains $(\leq nR)(x)$ and $R(x,y_1), \ldots, R(x,y_{n+1})$<br>such that $y_1, \ldots, y_{n+1}$ are distinct individuals<br>then for each pair $y_i, y_j$ $(i, j \in \{1, \ldots, n+1\}, i \neq j)$<br>$\mathscr{A}_{i,j}$ is obtained from $\mathscr{A}$ by replacing each occurrence of $y_i$ by $y_j$ |
| Rule $\forall$ | if $\mathscr{A}$ contains $(\forall R.C)(x)$ and $R(x,y)$, but not $C(y)$<br>then $\mathscr{A}' := \mathscr{A} \cup \{C(y)\}$ |

$\mathscr{I}$ of $\mathscr{T}$ such that $C^{\mathscr{I}} \neq \varnothing$. A concept $D$ *subsumes* $C$ towards $\mathscr{T}$ if $C^{\mathscr{I}} \subseteq D^{\mathscr{I}}$ holds for all models of $\mathscr{T}$.

Subsumption can be easily reduced to satisfiability: $C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable. Similarly, satisfiability can be reduced to subsumption: $C$ is satisfiable iff there is no concept name $A$ such that $C \sqsubseteq A \sqcap \neg A$. Satisfiability of concept descriptions can be reduced to the consistency problem: $C$ is satisfiable iff for some $a$ the ABox $\{C(a)\}$ is consistent.

One of the most prominent techniques for carrying out inferences in the framework of Description Logics is a variant of a *tableau algorithm* (see Baader and Sattler, 2001, for an overview). In the following, we present the main idea of a tableau algorithm for $\mathscr{ALCN}$ DL considered in this book.

### *Tableau Algorithm*

Given a concept description $C$, the satisfiability tableau algorithm constructs a minimal interpretation $\mathscr{I}$ that satisfies $C$. It is usually assumed that all concept descriptions in the corresponding ontology are in *negation normal form* (negation occurs only directly in front of concept names). The algorithm instantiates the ABox $\mathscr{A} = \{C(x)\}$ and applies consistency preserving expansion rules to this ABox as shown in Table 4.2.

Rule $\sqcup$ is nondeterministic in the sense that $\mathscr{A}$ is transformed into two new ABoxes such that the original ABox is consistent iff one of the new ABoxes is consistent. Similarly, at-most restrictions are treated nondeterministically by identifying some of the role successors.[19] Thus, the algorithm constructs not one ABox, but a finite set of ABoxes.

At-least and existential restrictions are treated by generating the required role successors as new individuals. Value restrictions impose new constraints on already existing role successors.

An ABox $\mathscr{A}$ is called *complete* iff none of the expansion rules of Table 4.2 applies to it. The ABox $\mathscr{A}$ contains a *clash* iff

a) $\{C(x), \neg C(x)\} \in \mathscr{A}$,

b) $\{\leq nR(x), \geq mR(x)\} \in \mathscr{A}, n > m$, or

c) $\{\bot(x)\} \in \mathscr{A}$.

An ABox is called *closed* if it contains a clash, and *open* otherwise. The satisfiability algorithm returns "satisfiable" if the set of ABoxes constructed using the expansion rules from Table 4.2 contains an open ABox; otherwise, "unsatisfiable" is returned. For correctness of the algorithm, see Baader and Sattler (2001). Satisfiability of $\mathscr{ALCN}$-concept descriptions is decidable and PSPACE-complete, see Baader and Sattler (2001).

The satisfiability algorithm described above can also be used to decide consistency of ABoxes. If $\mathscr{A}_0$ is an ABox then, in order to test it for consistency, one has to apply the expansion rules from Table 4.2 to the set $\mathscr{A}_0$.[20] Consistency of $\mathscr{ALCN}$-ABoxes is PSPACE-complete, cf. Baader and Sattler (2001).

Several reasoning systems implement Description Logic inferences.[21] In the study described in this book, we employ two different reasoners, which are *KAON2* and *HermiT*. *KAON2*[22] (Motik, 2006) is a free reasoner for $\mathscr{SHIQ}$-DL extended with a fragment of the Semantic Web Rule Language.[23] It implements a resolution-based decision procedure for general TBoxes (subsumption, satisfiability, classification) and ABoxes (retrieval, conjunctive query answering). The *HermiT* system[24] (Motik *et al.*, 2009) is a free reasoner for OWL 2/$\mathscr{SROIQ}$ with OWL 2 datatype support and support for description graphs.

[19] If $(a,b) \in R^{\mathscr{I}}$ then $b$ is called *R-successor* of $a$.

[20] Baader and Sattler (2001) point out that the algorithm obtained this way need not terminate. However, termination can be easily gained by requiring that generating rules (Rules $\exists$ and $\geq$) may only be applied if none of the other rules is applicable (Baader and Sattler, 2001).

[21] A list of the available reasoning tools can be found, for example, at *http://www.cs.man.ac.uk/~sattler/reasoners.html*

[22] *http://kaon2.semanticweb.org/*

[23] *www.w3.org/Submission/SWRL/*

[24] *http://www.comlab.ox.ac.uk/projects/HermiT/index.html*

It implements a hypertableau-based decision procedure, uses the OWL API 3.0, and is compatible with the OWLReasoner interface of the OWL API.

## 4.5 Concluding Remarks

This chapter describes mechanisms and tools underlying inference-based natural language understanding. Section 4.1 introduces two semantic parsers employed. The two parsers have a similar purpose – construction of logical forms for discourse fragments in English, but they produce structures of different complexity. The *ESG* parser outputs logical forms, which do not explicitly represent logical features of natural languages, such as, for example, quantification scope. In contrast, the *Boxer* system focuses on quantification and logical connectors in natural language. In the study described in this book, *Boxer*'s output is used in combination with a deductive theorem prover (see Sec. 8.2), while flat *ESG* output constitutes an input for an abductive reasoner (see Sec. 8.3).

Sections 4.2 and 4.3 describe the two principal reasoning mechanisms used in natural language understanding: deduction and abduction. Section 4.4 concentrates on a special case of deductive reasoning and introduces a decision algorithm for Description Logics. Abduction is often considered to be similar to deductive model building. Indeed, construction of interpretations in an abductive framework can be seen as model building. However, abductive and deductive reasoning have significant differences, which influence the resulting interpretations/models. These differences mainly concern treatment of axiom weights and proposition merging strategies. The following points are worth mentioning in this respect.

**Ambiguity** Deductive model builders have no means for distinguishing between alternative readings of a text fragment if all these readings are logically consistent with the rest of the discourse. Consider again the example *John composed the sonata* with two possible readings of the verb *compose*. Suppose the following deductive axioms are given

(1) $\forall e,x_1,x_2(compose(e,x_1,x_2) \rightarrow put\_together(e,x_1,x_2) \vee (create\_art(e,x_1,x_2) \wedge work\_of\_art(x_2)))$

(2) $\forall x(sonata(x) \rightarrow work\_of\_art(x))$

A deductive model builder will construct the following two models.

$\mathcal{M}_1 = \{John(j), compose(c,j,s), put\_together(c,j,s), sonata(s), work\_of\_art(s)\}$
$\mathcal{M}_2 = \{John(j), compose(c,j,s), create\_art(c,j,s), sonata(s), work\_of\_art(s)\}$

$\mathcal{M}_1$ and $\mathcal{M}_2$ are both consistent and have equal size. Thus, there is no possibility to distinguish between them. Therefore, NLU frameworks employing deduction use a disambiguation module based on statistics instead of reasoning, which precedes the inference module (cf. Sec. 8.2). In contrast, abductive reasoning allows us to select those readings, which are more coherent with the rest of the discourse and make the discourse redundancy explicit (cf. Sec. 4.3).

**Model minimality** Deductive reasoning allows us to assign the same individual to all variables in the model, unless some of the assignments are blocked by inconsistency of the model. This implies that disjointness axioms should be added for all incompatible predicates. Obviously, adding disjointness axioms substantially increases the size of the knowledge base and can cause slower reasoning. Given the sentence *John eats an apple and Bill eats an apple* with the logical form

$$\exists x_1,x_2,x_3,x_4,e_1,e_2(John(x_1) \wedge eat(e_1,x_1,x_2) \wedge apple(x_2) \wedge Bill(x_3) \wedge eat(e_2,x_3,x_4) \wedge apple(x_4)),$$

the axiom $\forall x(John(x) \rightarrow \neg Bill(x))$ will help to reject the model

$$\mathcal{M}_1 = \{John(j), Bill(j), eat(e,j,a), apple(a)\},$$

which states that John and Bill is the same person. However, there are no means to reject the following model:

$$\mathcal{M}_2 = \{John(j), Bill(b), eat(e_1,j,a), eat(e_2,b,a), apple(a)\},$$

which states that John and Bill eat the same apple. Thus, the concept of model minimality employed by most of the existing model builders seems to be inadequate for representing discourse models.

Weighted abduction also runs into this problem. The sentence above has the following flat logical form

$$John(x_1) \wedge eat(e_1,x_1,x_2) \wedge apple(x_2) \wedge Bill(x_3) \wedge eat(e_2,x_3,x_4) \wedge apple(x_4).$$

The propositions $apple(x_2)$ and $apple(x_4)$ have the same predicate name, therefore they should be unified, which implies that John and Bill eat the same apple. In order to avoid undesirable mergings, practical implementation of weighted abduction introduces non-merge constraints stating, which variables cannot be merged, see Sec. 7 for more details.

**Unbounded inference** An additional issue concerns the problem of where to stop drawing inferences, which could easily be unlimited in number. If the KB would contain axioms stating that John is a person, every person is a living being, every living being is a physical object, etc., then additional propositions ($living\_being(j), physical\_object(j), \ldots$) would be included into the deductive models constructed for the sentence *John composed the sonata*. However, it is rather questionable whether this information really belongs to the interpretation of the sentence. Moreover, a longer inference chain implies longer processing times. An abductive procedure does not run into this problem, because it naturally restricts possible inferences: An inference is appropriate if it is part of the lowest-cost proof of the logical form.

**Incomplete knowledge** Deductive reasoners have another shortcoming, namely their inability to reason given an incomplete knowledge base. In the cases when it is impossible to provide it with *all* the knowledge, which is relevant for interpretation of a particular piece of text, deductive reasoners may fail to find a prove. In practice, this might happen quite often (cf. Sec. 8.2). In turn, abduction solves the problem of knowledge incompleteness by allowing assumptions.

**Logical structure of natural language** As mentioned in Sec. 4.3, weighted abduction operates on flat formulas and has no special account for predicates representing logical connectors and quantification in natural language such as *if*, *not*, *or*, etc. Suppose the sentence *If Mary gives John a book, then he reads it* has the following FOL logical form

$$\forall e_1, x_1, x_2, x_3((Mary(x_1) \wedge give(e_1, x_1, x_2, x_3) \wedge book(x_2) \wedge John(x_3)) \rightarrow \\ \exists e_2(read(e_2, x_3, x_2))).$$

The corresponding flat logical form treated by weighted abduction looks as follows:

$$Mary(x_1) \wedge give(e_1, x_1, x_2, x_3) \wedge book(x_2) \wedge John(x_3) \wedge read(e_2, x_3, x_2) \wedge imply(e_1, e_2),$$

where the predicate *imply* stands for implication. Classical abductive approach suggests axiomatization of such predicates as *imply*, see Hobbs (1985b). However, this axiomatization has never been implemented, and one might doubt whether it would efficiently work in practice. If no axiomatization of this predicate is contained in the knowledge base, then the incorrect inferences like *Mary gives John a book* and *John reads a book* will follow from the logical form above. In the deductive framework, axiomatization of logical connectors and quantifiers is not required, because it follows from the calculus.

## Chapter 5

# Knowledge Base Construction

As already mentioned in previous chapters of this book, natural language understanding crucially depends on the underlying background knowledge. One of the central issues in this book is the construction of a knowledge base (KB) appropriate for natural language inferences. Instead of constructing the knowledge base from scratch, we propose to use existing knowledge resources freely available to the community. Chapter 3 gives a detailed overview of such resources, classifying them into lexical-semantic resources and ontologies. In Chap. 3, we conclude that both types of knowledge, being for the most part disjoint, are useful for knowledge-intensive NLP.

The proposed integrative knowledge base is organized in a modular way because of the following two reasons:

(1) Substitutability of modules
Lexical-semantic knowledge depends on a target language, while ontological knowledge is mostly language-independent. Existing lexical-semantic databases are for the most part domain-independent; in contrast, ontologies may be strongly focused on a particular domain. Thus, keeping lexical and ontological knowledge apart enables easier switching to another language or application domain.

(2) Different reasoning mechanisms
As explained in Chap. 3, lexical-semantic knowledge is for the most part fuzzy, probabilistic, and non-definitional. Therefore, an inference procedure intended to handle this type of knowledge must tolerate inconsistencies and borderline cases, and it must rely on probabilistic information. In contrast, ontologies are usually explicitly intended to be logically consistent and unambiguous. Moreover, ontologies usually require an expressive representation language for their formalization, while lexical-semantic knowledge is for the most part structurally simple. The modular organization of the knowl-

edge base enables us to use different reasoning mechanisms for the different types of knowledge as it is shown in Chap. 7.

The proposed integrative knowledge base consists of three modules, which are shown in Fig. 5.1. The lexical-semantic module contains axioms, which were extracted from lexical-semantic resources and enriched with probabilistic knowledge automatically mined from corpora. This module is language-dependent and domain-independent. Lexical-semantic knowledge is linked to an ontological module via a lexicon-ontology mapping. The ontological module contains ontologies axiomatizing particular domains or general-purpose ontologies. The distributional module contains unstructured dependencies between words mined from corpora. This knowledge is intended to reveal dependencies, which cannot be inferred on the basis of the axioms. It is language-dependent and can be both domain-specific (if learned from domain-specific corpora) or domain-independent (if learned from general-purpose corpora). The modules of the proposed KB are stored and queried separately (cf. Chap. 7). Lexical-semantic knowledge is processed in the framework of weighted abduction, while ontological knowledge is queried by a Description Logic (DL) reasoner.

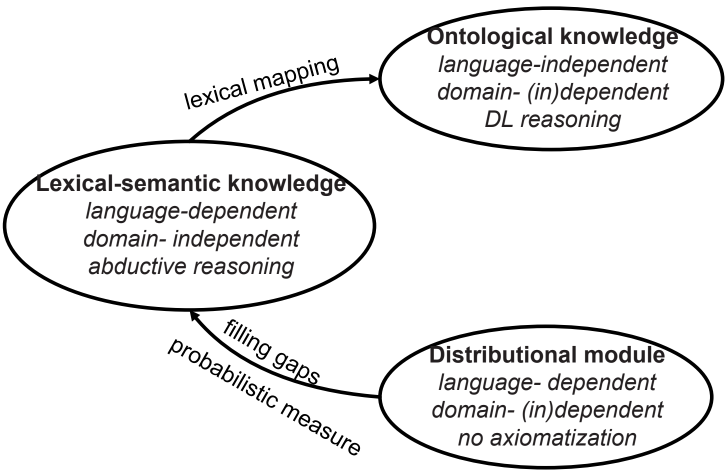


Fig. 5.1 Modules of the proposed integrative knowledge base.

The present chapter introduces the integrative knowledge base constructed in the framework of the study described in this book. We did not intend to exploit all available knowledge resources, because there is definitely much more knowledge available than any reasoning system can handle. Our goal was rather to show how some of the large and well-developed knowledge resources can be turned into axioms applicable to natural language understanding and to demonstrate that the proposed strategy has advantages and potentials.

## 5.1 Preliminaries

### 5.1.1 *Notation*

Before starting to document generated axioms, we explain our notation. Logical forms are given here in the style of Hobbs (1985b). They are conjunction of propositions, in which all variables are existentially quantified with the widest possible scope. Hobbs (1985b) extends Davidson's approach (Davidson, 1967) to all predications and claims that corresponding to any predication that can be made in natural language, there is an eventuality. Correspondingly, any predication in the logical notation has an extra argument, which refers to the "condition", in which that predication is true. Thus, in the logical form $John(e_1, j) \wedge run(e_2, j)$ for the sentence *John runs*, $e_2$ is a running event by John (John's running) and $e_1$ is a condition of $j$ being named "John". Corresponding to every $n$-ary predicate $p$, Hobbs (1985b) introduces an $n$+1-ary predicate $p'$ such that its first argument refers to the condition that holds when $p$ is true of the subsequent arguments.

In the axioms described below, all verb predicates have four arguments such that the second argument is reserved for a prototypical agent, the third argument is a prototypical theme/patient, and the forth argument stands for a prototypical goal/recipient (see Dowty, 1991, for a description of the prototypical roles). Even if some arguments are not instantiated in the discourse, they are present in the logical form as "dummies". For example, the sentence *John runs* will have the following logical form: $John(e_1,x_1) \wedge run(e_2,x_1,u_1,u_2)$.

As already mentioned in Sec. 4.3, abductive axioms should be read "right to left". Given the axiom $p(x_1,x_2) \rightarrow q(x_1,y_1)$, the existence of the proposition $q(x_1,y_1)$ on the right hand side implies the assumption of the existence of the proposition $p(x_1,x_2)$ on the left hand side, taking into account identity of the first argument of $p$ and the first argument of $q$, which is $x_1$.

### 5.1.2 *Axiom Weights*

As described in Sec. 4.3, each proposition on the left hand side of an abductive axiom should be assigned a weight, which defines the cost of the assumption of this proposition.

A cost function $f$ is applied in order to calculate the assumption cost. In the proposed framework, the cost function $f(w,c) = w \cdot c$ is used, such that $w$ is the assumption weight and $c$ is the cost of the input proposition. For example, given the axiom $P^w \rightarrow Q$ and the input proposition $Q$ with the cost $c$, the assumption of $P$ costs $w \cdot c$.

Axiom weights are calculated using frequency of the corresponding word senses and semantic relations, which is derived from corpora as described in the next sections of this chapter. In our framework, axioms of the type *species* $\rightarrow$ *genus* should have costs exceeding 1, which means that assuming *species* costs more than assuming *genus*, because there might be many possible *species* for the same *genus*. The costs of such axioms are heuristically defined as ranging from 1 to 2.

The frequency-based method allows us to calculate one total weight for an axiom. If the axiom contains more than one proposition on the left hand side, then the total weight is equally divided between the propositions. For example, given the axiom $P_1 \wedge ... \wedge P_n \rightarrow Q$ and the weight $w$ calculated for this axiom, the weight $w/n$ is assigned to each $P_i$ $(1 \leq i \leq n)$.

## 5.2 Axioms derived from Lexical-Semantic Resources

As stated in Sec. 3.1.1, WordNet (Fellbaum, 1998b) and FrameNet (Ruppenhofer *et al.*, 2010) are currently the semantically richest lexical-semantic resources available. The process of generating axioms for reasoning from both of these resources is described in Sec. 5.2.1 and 5.2.2. Moreover, we experimented with an automatically generated lexical-semantic Proposition Store (Peñas and Hovy, 2010), which is considered in Sec. 5.2.3. Section 5.3 presents the ontology employed for the interpretation of domain-specific texts. Section 5.4 is devoted to the construction of the distributional similarity space used to infer relationships, which are not recoverable on the basis of the axioms in the knowledge base.

### 5.2.1 *Axioms derived from WordNet*

WordNet provides two types of information: a) mappings from lexemes to word senses represented by synsets and b) relations defined on word senses (cf. Sec. 3.1.1). The resource mostly focuses on paradigmatic relations such as synonymy, hypernymy, and meronymy. Thus, WordNet strongly supports hierarchical reasoning which is the backbone of the most valid inferences. For example, WordNet contains the hypernymy relation between the synsets $S_1 = \{$*dog*, *domestic dog*, *Canis familiaris*$\}$ and $S_2 = \{$*animal*, *animate being*, *beast*, *brute*, *creature*, *fauna*$\}$. Suppose there are two sentences *John bought a dog* and *John bought an animal*. If an NLU system manages to map *dog* in the first sentence to $S_1$ and *animal* in the second sentence to $S_2$ then the inference that the first sentence entails the second one seems to be straightforward. The problem is that mapping a lexeme to an appropriate synset is far from being trivial, because each lexeme can participate in many different synsets. For example, *dog* is also a member of the synset $S_3 = \{frump, dog\}$ (a dull unattractive unpleasant girl or woman), which has the synset $\{person, \ldots\}$ rather than $\{animal, \ldots\}$ as its hypernym. Thus, in order to achieve correct inferences one needs a disambiguation procedure.

Different solutions have been proposed so far for the disambiguation into WordNet synsets. The most obvious solution is to choose the most frequent sense.[1] Since *dog*$\rightarrow S_1$ is much more frequent than *dog*$\rightarrow S_3$ in the example above, a disambiguation procedure based on this solution will always assign $S_1$ to each occurrence of *dog*. This strategy was realized, for example, by Bos and Markert (2005).

Another solution concerns using statistical disambiguation tools trained on WordNet-annotated corpora. This method is more resource-consuming in comparison to the first one. Unfortunately, it does not really give a large improvement in performance as compared to the strategy of choosing most frequent senses, see Palmer *et al.* (2006) and Sec. 8.2 for more details.

We propose to employ weighted abduction for disambiguation. Instead of applying a separate disambiguation procedure before reasoning, disambiguation is done during the inference process so that discourse redundancy is exploited (cf. the "sonata"-example in Sec. 4.3). The lexeme-synset mappings corresponding to word senses are used to generate disambiguation axioms. For example, in the abductive axiom below, the verb *compose* is mapped to synset 123 in WordNet. Synset propositions have two arguments ($e_0$ and $e_1$ in the example below). The first argument refers to the "condition", in which this proposition is true, i.e. in which the word referred to by the second argument is a part of the given synset. The second argument corresponds to the first argument of the word sense proposition on the right hand side.

$$synset\text{-}123(e_0, e_1) \rightarrow compose(e_1, x_1, x_2, u_1)$$

These axioms need to be weighted. In order to assign a weight $w_i^j$ to a sense $i$ of a lexeme with an index $j$, we use information about the frequency $f_i^j$ of the word sense in WordNet-annotated corpora. An obvious way of converting the frequency $f_i^j$ to the weight $w_i^j$ would be the following equation:

[1] Recall that WordNet provides frequency of every word sense in annotated corpora.

$$w_i^j = 2 - \frac{f_i^j}{\sum\limits_{1 \leq n \leq |S_j|} f_n^j}, \tag{5.1}$$

where $S_j$ is a set of all senses of the lexeme with index $j$.[2] Suppose a lexeme with the index 1 has two senses with the frequencies $f_1^1 = 1$, $f_2^1 = 10$. According to Eq. 5.1, the corresponding axioms will be assigned the following weights: $w_1^1 \approx 1.9$, $w_2^1 \approx 1.09$. Thus, a more reliable mapping receives the lower weight, which results in a lower cost of assuming a more frequent word sense.

Moreover, the following WordNet relations defined on synsets were converted into axioms: hypernymy, instantiation, entailment, causation, similarity, meronymy (cf. Sec. 3.1.1). Hypernymy and instantiation relations presuppose that the related synsets refer to the same entity and condition (the first axiom below), whereas other types of relations relate synsets referring to different entities (the second axiom below). All axioms based on WordNet synset relations have weights equal to 1.5.

$synset\text{-}1(e_1, e_0) \rightarrow synset\text{-}2(e_2, e_0)$
$synset\text{-}1(e_1, e_2) \rightarrow synset\text{-}2(e_2, e_4)$

The direction of the entailment for every synset relation is defined in Table 5.1, where $\rightarrow$ stands for logical implication. Concerning meronymy, defining the entailment direction is not trivial, because both directions are possible. For example, every building has a wall as its part, but the inverse is not necessary (although typically) true. At the same time, every nose (the organ of smell) presupposes existence of a living being, to which this nose belongs, but not every living being has a nose. In order to convert meronymy relations from WordNet into axioms properly, an additional annotation of the entailment direction is required. Since WordNet does not provide this annotation, we introduce two axioms for each meronymic relation, which represent both entailment directions.

WordNet also provides morphosemantic relations which relate verbs and nouns, e.g., *buy-buyer*. WordNet distinguishes between 14 types of such relations.[3] We use relation types in order to define the direction of the entailment and map the proposition arguments, see Table 5.2. Axioms based on WordNet morphosemantic relations have weights equal to 1. Unfortunately, the relation types are not explicitly defined by the resource developers, therefore the axiomatization was done mostly on the basis of examples.

[2]Recall that the axioms corresponding to "unsafe" assumptions should have weights ranging from 1 to 2.
[3]*http://wordnet.princeton.edu/wordnet/download/standoff/*

Table 5.1 Direction of entailment for WordNet relations.

| **Relation** | **Direction of entailment** | **Example** |
|---|---|---|
| Hypernymy | hyponym → hypernym | *dog* → *animal* |
| Instantiation | instance → class | *White House* → *residence* |
| Entailment | event → entailed event | *snore* → *sleep* |
| Causation | cause → effect | *kill* → *die* |
| Similarity | $\text{concept}_1 \leftrightarrow \text{concept}_2$ | *wet* ↔ *watery* |
| Meronymy | whole ↔ part | *tree* ↔ *trunk* |

Given the relations "agent" and "body-part", existence of the verb implies that the corresponding noun can be used for referring to the second argument (agent) of the verb, e.g., if there is a situation, in which X buys Y, then there is a buyer X. Relations relating a verb to a verbal noun ("event" and "state") are bidirectional and imply that the noun refers to the verb's conditions of being true, i.e. the first argument of the verb is equal to the second argument of the noun. For example, *arrive* and *arrival* point to the same event. Concerning all other relations, the verb refers to an event, which presupposes an existence of an object denoted by the noun, which describes an instrument, a location, a result, etc. of this event. For some of the relations ("agent", "undergoer", "destination"), the noun denotes an event's role, which necessarily implies that the event took place, e.g., the existence of an addressee implies that there has been an addressing event.

Additionally, we exploited the WordNet synset definitions. In WordNet, the definitions are given in a natural language form. For example, the synset {*horseback*, *ahorse*, *ahorseback*} is defined as *on the back of a horse*. We used the extended WordNet resource[4], which provides logical forms for the definition in WordNet version 2.0 (Harabagiu *et al.*, 1999). For example, the definition for the synset containing such lexemes as *horseback* has the following form in extended WordNet[5]:

```
on:IN(e1, x2) back:NN(x2) of:IN(x2, x1) horse:NN(x1)
```

In this logical form, the variable `e1` refers to an unspecified predicate, to which the prepositional phase can be attached in a longer text fragment, e.g., *to* ***sit*** *on the back of a*

[4] *http://xwn.hlt.utdallas.edu/*
[5] Note that this logical form does not contain articles, which are omitted on purpose.

Table 5.2 Axioms derived from WordNet morphosemantic relations.

| **Relation type** | **Axiomatization** | **Example** |
|---|---|---|
| agent | $verb(e_0,\underline{\mathbf{x_1}},x_2,x_3) \leftrightarrow noun(e_1,\underline{\mathbf{x_1}})$ | *buy-buyer* |
| body-part | $verb(e_0,\underline{\mathbf{x_1}},x_2,x_3) \rightarrow noun(e_1,\underline{\mathbf{x_1}})$ | *abduct-abductor* |
| event | $verb(\underline{\mathbf{e_0}},x_1,x_2,x_3) \leftrightarrow noun(e_1,\underline{\mathbf{e_0}})$ | *arrive-arrival* |
| state | $verb(\underline{\mathbf{e_0}},x_1,x_2,x_3) \leftrightarrow noun(e_1,\underline{\mathbf{e_0}})$ | *approve-approval* |
| undergoer | $verb(e_0,x_1,x_2,x_3) \leftrightarrow noun(e_1,x_0)$ | *address-addressee* |
| destination | $verb(e_0,x_1,x_2,x_3) \leftrightarrow noun(e_1,x_0)$ | *patent-patentee* |
| material | $verb(e_0,x_1,x_2,x_3) \rightarrow noun(e_1,x_0)$ | *insulate-insulator* |
| instrument | $verb(e_0,x_1,x_2,x_3) \rightarrow noun(e_1,x_0)$ | *poke-poker* |
| by-means-of | $verb(e_0,x_1,x_2,x_3) \rightarrow noun(e_1,x_0)$ | *dilate-dilator* |
| vehicle | $verb(e_0,x_1,x_2,x_3) \rightarrow noun(e_1,x_0)$ | *kayak-kayak* |
| location | $verb(e_0,x_1,x_2,x_3) \rightarrow noun(e_1,x_0)$ | *bath-bath* |
| result | $verb(e_0,x_1,x_2,x_3) \rightarrow noun(e_1,x_0)$ | *liquify-liquid* |
| property | $verb(e_0,x_1,x_2,x_3) \rightarrow noun(e_1,x_0)$ | *beautify-beauty* |
| uses | $verb(e_0,x_1,x_2,x_3) \rightarrow noun(e_1,x_0)$ | *classify-class* |

*horse* or *to **ride** on the back of a horse*. Other variables (`x1`, `x2`) refer to the arguments of the nouns *back* and *horse*, which are related with the preposition *of*.

Every logical form in extended WordNet is annotated as `GOLD`, `SILVER`, or `NORMAL`. It is `GOLD` if a human decided over that information; it is `SILVER` if there was agreement in an automatic voting process, but no human evaluation; it is `NORMAL` if output was provided by the software without human evaluation or voting agreement.

We converted logical forms from extended WordNet into axioms adding "condition" arguments to every proposition. For example, the following axiom represents the meaning of the synset containing such lexemes as *horseback*, where $e_0,e_2,e_3,e_4$ are "condition" arguments.

$$on(e_0,e_1,x_2) \wedge back(e_2,x_2) \wedge of(e_3,x_2,x_1) \wedge horse(e_4,x_1) \rightarrow synset\text{-}X(e_5,e_0)$$

The axiom above implies that if a lexeme in the logical form was mapped to a word sense participating in *synset-X*, then the input logical form can be extended with the definition of the synset. The synset definition might introduce propositions revealing redundancy. For example, given the text fragment *Alexander was riding on the back of Bucephalus. He liked to appear ahorse*, the axiom above will help to link both sentences.

Table 5.3 Statistics for axioms extracted from WordNet.

| **Axiom type** | **Source** | **Number of axioms** |
|---|---|---|
| Lexeme-synset mappings | WN 3.0 | 207 000 |
| Lexeme-synset mappings | WN 2.0 | 203 100 |
| Synset relations | WN 3.0 | 141 000 |
| Derivational relations | WN 3.0 (annotated) | 35 000 |
| Synset definitions | WN 2.0 (parsed, annotated) | 115 400 |

Axiom weights were assigned according to the quality of the annotation. Axioms resulting from `GOLD` annotations have weights equal to 1. `SILVER` and `NORMAL` axioms have weights of 1.2 and 1.4 respectively.

Statistics of the axioms extracted from WordNet is given in Table 5.3. The number of axioms is approximated to the nearest hundred.

### 5.2.2 *Axioms derived from FrameNet*

The previous attempts of using FrameNet (FN) for reasoning tasks mostly consider the paraphrasing task, which consists in matching FN-annotated text fragments. Such matching becomes nontrivial if the text fragments are annotated with different frames. Considering the example below, in order to match *at least 11 people* in (a) with *Humans* in (b) and *An avalanche* in (a) with *in an avalanche* in (b) one needs to use the semantic relation between KILLING and DEATH.[6]

(a) [*An avalanche*]$_{\text{CAUSE}}$ *has struck a popular skiing resort in Austria*, [*killing*]$_{\text{KILLING}}$ [*at least 11 people*]$_{\text{VICTIM}}$.

(b) [*Humans*]$_{\text{PROTAGONIST}}$ [*died*]$_{\text{DEATH}}$ [*in an avalanche*]$_{\text{CAUSE}}$.

Since FrameNet contains the causation relation linking KILLING and DEATH, such that VICTIM in KILLING is related to PROTAGONIST in DEATH and CAUSE in KILLING is related to CAUSE in DEATH, the matching becomes possible.[7]

[6] In the framework of FrameNet a predicate in a predicate-argument construction is annotated with a frame name (e.g., KILLING, DEATH) and other elements are annotated with some frame roles of that frame (e.g., CAUSE, VICTIM, PROTAGONIST).

[7] Here, we do not consider aspects related to matching role fillers, which have different lexical and syntactic realizations. In applications, the reader will find several solutions, such as using WordNet, extracting syntactic heads, or using statistical similarity measures.

Following the strategy sketched above, Shen and Lapata (2007) make use of the inheritance relation in FN for question answering. Scheffczyk *et al.* (2006) propose a reasoning procedure involving all types of FN relations. A detailed overview of applications of the frame matching to recognizing textual entailment is given in Burchardt *et al.* (2009).

Approaches employing FrameNet are concerned with the disambiguation problem, because in FrameNet, like in WordNet, one lexeme can be mapped to several senses evoking different frames. Most of the approaches rely on statistical disambiguation tools trained on corpora manually annotated with FrameNet frames and roles, see Erk and Pado (2006); Das *et al.* (2010). Since the size of FN-corpora is quite limited, the quality of the automatic frame assignment is limited as well. We propose to derive axioms for lexeme-frame mappings and to apply weighted abduction for disambiguation. The lexeme-frame mappings are based on syntactic patterns contained in FrameNet as described below.

Let us introduce the notation for frame axiomatization. In our framework, every frame is represented by a proposition of the form $\text{FRAME\_NAME}(e_0, x_1, \ldots, x_n)$, where FRAME_NAME is the name of the frame in FrameNet, $e_0$ is the argument referring to the condition, in which the proposition is true, and $x_1, \ldots, x_n$ are arguments corresponding to the fillers of the frame roles. For each frame role, an argument position is reserved. For example, the fillers of the CAUSE role of the KILLING frame will always be referred to by the second argument $x_1$ of the proposition $\text{KILLING}(e_0, x_1, \ldots, x_n)$. In case there is a need to represent the name of the frame role, this is achieved by introducing a proposition of the form $\text{ROLE\_NAME}(e_1, e_0, x_1)$, where $e_0$ is the "condition" argument of the corresponding frame, $x_1$ refers to a filler of the role, and $e_1$ is the argument referring to the condition, in which the proposition is true, for example, $\text{KILLING}(e_0, x_1, x_2) \wedge \text{CAUSE}(e_1, e_0, x_1)$.

#### 5.2.2.1 *Axioms Mapping Predicate-argument Structures to Frames*

As described in Sec. 3.1.1, FrameNet, similar to WordNet, maps lexemes to clusters of word senses (frames) and provides semantic relations between these clusters (frame relations). However, frames are not atomic concepts; they have a structure consisting of frame roles. Correspondingly, predicate-argument structures rather than lexemes are mapped to frames and frame roles.

For example, FrameNet provides the following two predicate-argument patterns showing how syntactic arguments of the verb *give* can be mapped to the semantic roles of the frame GIVING:

| DONOR | RECIPIENT | THEME | Frequency |
|---|---|---|---|
| `NP Ext` | `NP Dep` | `NP Obj` | 9 |
| `NP Ext` | `PP[to] Dep` | `NP Obj` | 5 |

In this example, `NP Ext` (external argument) stands for a syntactic subject, `NP Obj` (object) is a direct object, and `NP Dep` (dependent) is an indirect object. The first pattern above corresponds to the constructions like [*John*]$_{\text{DONOR}}$ [*gave*]$_{\text{GIVING}}$ [*Mary*]$_{\text{RECIPIENT}}$ [*the book*]$_{\text{THEME}}$. The second pattern – less frequent – is derived from constructions like [*John*]$_{\text{DONOR}}$ [*gave*]$_{\text{GIVING}}$ [*the book*]$_{\text{THEME}}$ [*to Mary*]$_{\text{RECIPIENT}}$. The RECIPIENT role in this pattern is filled by a prepositional phrase. Frequencies of these patters show how many times the patterns have occurred in the FrameNet-annotated corpora.

We use FrameNet syntactic patterns in order to derive axioms. Axiom weights are calculated on the basis of pattern frequencies as shown in Eq. 5.1. For example, the following axioms correspond to the patterns above:

$$\text{GIVING}(e_0,x_1,x_2,x_3) \wedge \text{DONOR}(e_1,e_0,x_1) \wedge \text{RECIPIENT}(e_2,e_0,x_2) \wedge \text{THEME}(e_3,e_0,x_3)$$
$$\rightarrow give(e_0,x_1,x_2,x_3)$$
$$\text{GIVING}(e_0,x_1,x_2,x_3) \wedge \text{DONOR}(e_1,e_0,x_1) \wedge \text{RECIPIENT}(e_2,e_0,x_2) \wedge \text{THEME}(e_3,e_0,x_3)$$
$$\rightarrow give(e_0,x_1,u_1,x_3) \wedge to(e_4,e_0,x_2)$$

The first axiom above directly maps arguments $x_1,x_2,x_3$ of the *give* predicate to frame roles. In the second axiom, the variable $u_1$ is "dummy", because the recipient is syntactically expressed using the prepositional phrase with the preposition *to*. Therefore, the argument $x_2$ of *to* rather than $u_1$ is mapped to the RECIPIENT role.

FrameNet annotates frame role fillers with two markers: a phrase type (PT), e.g., `NP` or `PP`, and a grammatical function (GF), e.g., `Ext` or `Dep`.[8] The main phrase types are noun phrases ([*John*] ***runs***, [*John's*] ***arrival***, [*car*] ***manufacturer***), prepositional phrases (***go*** [*to work*], ***peeled*** *the skin* [*off*], ***think*** [*about going home*]), verb phrases (***make*** *him* [*go*], ***enjoy*** [*reading*]), clauses (***tell*** *me* [*where is John*], *John* ***likes*** [*Mary to read*]), adjective phrases ([*economic*] ***policy***, ***turn*** [*red*]), and adverb phrases ([*greatly*] ***reduced***).

Grammatical functions in FrameNet describe not the surface-syntactic positions of the role fillers, but the ways, in which the role fillers satisfy abstract grammatical requirements of the word evoking the frame. Consider the sentence *Circumstances forced the doctor to* ***treat*** *her enemies*[9] containing the word *treat*, which evokes the frame CURE. The word *circumstances* is the syntactic subject of this sentence, but it is not a semantic argument of

[8] The full list of markers can found in Ruppenhofer *et al.* (2010).
[9] This example is provided by Ruppenhofer *et al.* (2010).

*treat*. FrameNet annotates the NP *the doctor* as the external argument (`Ext`) of *treat*, even though it is not the surface subject of the sentence, because it satisfies a valence requirement of the verb *treat* outside the phrase headed by *treat* (therefore "external"). This treatment of grammatical functions is reasonable from the lexicographic point of view, but it makes translation of FrameNet patterns into axioms rather tricky. In the following, we describe the translation of patterns into logical forms for every part of speech.

*Verbs*

Role fillers of frames evoked by verbs can be annotated with the following GFs: External Argument (`Ext`), Object (`Obj`), and Dependent (`Dep`). External Arguments are outside of the maximal phrase headed by the frame evoking word and are linked functionally to the frame evoking word. There are the following four types of `Ext` in FrameNet:

(1) a subject of a finite verb evoking a frame, e.g., [*John*] ***runs***
(2) any constituent, which controls the subject of a frame evoking verb, e.g., [*The doctor*] *tried to* ***cure*** *me*, *They gestured* [*to us*] *to* ***leave***
(3) a dependent of a governing noun, e.g., [*the physician's*] *decision to* ***perform*** *the surgery*, *decision* [*by the Court*] *to* ***approve*** *our request*
(4) a passivized object in passive sentences, e.g., [*The book*] *was* ***given*** *to Mary*, [*Mary*] *was* ***given*** *a book*

In the cases of type (1), (2), and (3) above, the role filler marked with `Ext` can be mapped to the second argument of the corresponding proposition in the logical form. Patterns derived from passive sentences are ignored by the translation procedure, because FrameNet does not distinguish between passivized direct and indirect objects (cf. examples of type (4)); therefore, it is impossible to disambiguate passive patterns and map role fillers to frame roles properly.

The rule for converting `Ext` role fillers into arguments in logical forms is the following: If the `Ext` role filler is an NP or a clause, which is not passivized, then map it to the second argument of the corresponding proposition in the logical form. For example, in the axioms mapping the verb *give* to the frame GIVING above, $x_1$ is the second argument of the proposition *give* and it points to the role DONOR marked in FrameNet with `NP Ext`.

Object `Obj` is any normal direct object or any post-verb NP, which controls the subject of a complement of the frame evoking verb: *John* ***reads*** [*a book*], *They* ***expect*** [*us*] *to finish soon*, [*What*] *did you* ***cook*** *for dinner*. `Obj` role fillers are mapped to the third argument of the corresponding proposition in the logical form. For example, in the axioms for the verb *give* above, $x_2$ is the third argument of the proposition *give* and it points to the role THEME marked in FrameNet with `NP Obj`.

Dependent is defined by Ruppenhofer *et al.* (2010) as "the general grammatical function assigned to Adverbs, PPs, VPs, Clauses (and a small number of NPs), which occur after their governing verbs, adjectives or nouns in normal declarative sentences". Examples of the PP dependents are marked in the following sentences: *John* ***spoke*** [*to me*], *Bill* ***sold*** *the house* [*in order to finance a concert*]. PP dependents are mapped to propositions representing corresponding preposition. For example, in the second axiom for the verb *give* above, the preposition *to* relates the verb argument $e_0$ to the argument $x_2$ pointing to the RECIPIENT marked in FrameNet with `PP[to] Dep`.

Some NPs are marked as Dependents rather than as Objects if they are not passivizable and they express meanings normally associated with adjuncts and PP complements (e.g., place and time), for example, ***I run*** [*ten miles*] *every day*. Furthermore, the second object of ditransitives is treated as a Dependent, for example, in the sentence *They* ***gave*** [*the children*] *candy*. Since there is no possibility for distinguishing between these cases, `NP Dep` role fillers are always mapped to the fourth argument of the corresponding proposition in the logical form. For example, in the first axiom for the verb *give* above, $x_3$ is the fourth argument of the proposition *give* and it points to the role THEME marked in FrameNet with `NP Dep`.

*Nouns*

FrameNet assigns the grammatical function `Dep` to any post-nominal complement of a frame evoking noun, for example, *the* ***fact*** [*that cats have fur*], *our* ***arrival*** [*at the station*]. The GF `Dep` is also assigned to any pre-nominal phrases (noun, adjective, gerund, or participle) that fill frame element roles, for example, [*medical*] ***book***, [*broken*] ***lamp***, [*allergy*] ***treatment***. Some dependents of nouns are realized as the predicates of copular sentences, for example, *the* ***fact*** *is* [*that cats have fur*].

The translation procedure maps PP dependents of nouns to arguments of prepositions in the logical form. For example, the noun *arrival* evoking the frame ARRIVING has the pattern GOAL: `PP[at] Dep` in FrameNet. The following axiom is derived from this pattern:

$\text{ARRIVING}(e_0,\ldots,x_1,\ldots) \wedge \text{GOAL}(e_1,e_0,x_1) \rightarrow arriving(e_0,x_2) \wedge at(e_2,x_2,x_1)$[10]

Noun dependents are mapped using the noun compound predicate *nn*. For example, given the pattern AFFLICTION: `N Dep` defined for the noun *treatment* evoking the CURE frame (e.g., [*cancer*] ***treatment***), the following axiom is produced:

$\text{CURE}(e_0,\ldots,x_2,\ldots) \wedge \text{AFFLICTION}(e_1,e_0,x_2) \rightarrow treatment(e_0,x_1) \wedge nn(e_2,x_2,x_1)$

The GF `Gen` is assigned to any possessive NP functioning as determiner of a frame evoking noun, e.g., [*Shakespeare's*] ***book***. Such patterns are translated using possessive predicate *poss* in logical forms, for example:

$\text{TEXT}(e_0,\ldots,x_2,\ldots) \wedge \text{AUTHOR}(e_1,e_0,x_2) \rightarrow book(e_0,x_1) \wedge poss(e_2,x_2,x_1)$

*Adjectives*

FrameNet has two different markers for the role fillers associated with a frame evoked by an adjective: `Ext` ([*the chair*] *is* ***red***) and `Head` (*the* ***small*** [*children*]). Both types of phrases are represented by the logical form $noun(e_1,x_1) \wedge adjective(e_2,x_1)$. Thus, both types of markers are translated into axioms as shown by the following example:

$\text{COLOR}(e_0,\ldots,x_1,\ldots) \wedge \text{ENTITY}(e_1,e_0,x_1) \rightarrow red(e_0,x_1)$

Some adjectives require prepositional frame role fillers, to which FrameNet assigns the GF `Dep`, e.g., *Lee is* ***certain*** [*of his innocence*]. `PP Dep` markers are translated into PP propositions as shown by the example below:

$\text{CERTAINTY}(e_0,\ldots,x_2,\ldots) \wedge \text{CONTENT}(e_1,e_0,x_2) \rightarrow certain(e_0,x_1) \wedge of(e_2,e_0,x_2)$

There are a few N complements of adjectives, which are marked in FrameNet with `N Head`, e.g., *suffering from* [*stupidity*] ***related*** *injuries*. In logical forms, such dependencies are represented by the *amod* predicate, for example, $stupidity(e_1,x_1) \wedge related(e_2,x_2) \wedge amod(e_3,e_1,e_2)$. A corresponding axiom example is shown below.

$\text{COGNITIVE\_CONNECTION}(e_0,\ldots,e_2,\ldots) \wedge \text{CONCEPT\_1}(e_1,e_0,e_2) \rightarrow$
$related(e_0,x_1) \wedge amod(e_3,e_2,e_0)$

*Adverbs*

Adverbial modifiers can modify an event or relation marked as `Head`, e.g., [*open the door*] ***carefully***, [*quite*] ***honestly***. Any element that modifies the frame evoking adverb

[10]The variable $x_1$ being an argument of the predicate ARRIVING occupies the argument position reserved in this predicate for the frame role GOAL.

receives the GF Dep. Unfortunately, such frame role fillers cannot be represented in logical forms, because there are no pointers from an adverb to its modifiers. For example, in the logical form $quite(e_1, e_2) \wedge honestly(e_2, e_3)$ the proposition *honestly* does not contain an argument pointing to *quite*.

*Prepositions*

FrameNet marks all possible dependents of prepositions with Dep, e.g., ***before*** [*the meal*], ***before*** [*returning to work*]. For example, given the phrase *John called me before returning to work* and the proposition *before*$(e_0, e_1, e_2)$ in the corresponding logical form, its second argument $e_1$ refers to the preceding event (*John called me*) and the third argument $e_2$ refers to the succeeding event (*returning to work*). Thus, translating preposition dependents into logical forms is straightforward:

$$\text{TIME\_VECTOR}(e_0, \ldots, e_3, \ldots) \wedge \text{LANDMARK\_EVENT}(e_1, e_0, e_3) \rightarrow before(e_0, e_2, e_3)$$

A constituent, which expresses an element of the frame associated with a preposition, which is outside the prepositional phrase, is tagged as Ext, e.g., [*John called me*] ***before*** *returning to work*. The Ext constituent is mapped to the second argument of the proposition in the logical form as shown below.

$$\text{TIME\_VECTOR}(e_0, \ldots, e_2, \ldots) \wedge \text{EVENT}(e_1, e_0, e_2) \rightarrow before(e_0, e_2, e_3)$$

*Phrases*

In FrameNet, not only single words, but also phrases are mapped to frames. For example, *play by the rules* evokes the frame COMPLIANCE, *chest of drawers* evokes CONTAINERS, and *down the road* evokes TIME_VECTOR. In order to constitute axioms, such phrases need to be converted into logical forms, for example:

$$\text{COMPLIANCE}(e_1, \ldots) \rightarrow play(e_1, x_1, x_2, u_1) \wedge by(e_2, e_1, x_4) \wedge rule(e_4, x_4)$$

5.2.2.2 *Axioms derived from Frame Relations*

FrameNet introduces semantic relations defined on frames such as inheritance, causation or precedence (see Table 3.2 for the full list of relations). For example, the GIVING and GETTING frames are connected with the causation relation. Roles of the connected frames are also linked, e.g., DONOR in GIVING is linked with SOURCE in GETTING.

Frame relations have no formal semantics in FrameNet. In order to generate corresponding axioms, a formalization of frame relations is required. In this section, we propose an axiomatization of frame relations; its ontological interpretation is discussed in Sec. 6.1.

Some frame relations imply relationships that go beyond entailment. For example, the precedence relation implies temporal precedence and the subframe relation implies spatial inclusion. These relations can be further linked to an appropriate ontological theory of time, space, or cause. Since investigating and testing these theories is out of the scope of this book, we will sometimes introduce blank undefined predicates which can be later linked to preferred theories.

As already mentioned in Sec. 3.1, FrameNet frames describe prototypical situations expressed by natural language expressions (see Sec. 6.1 for a deeper analysis). Thus, relations defined on frames should imply relations between situations, to which these frames refer to. In the following, we characterize the sort of axioms represented by frame relations using usual FOL notation and providing corresponding abductive axioms. For a frame $f$ being instantiated by a situation $s$ we will write $f(s)$.

*Inheritance*

The inheritance relation in FN is claimed to be similar to the ontological relation *is-a*. Informally speaking, "anything which is strictly true about the semantics of the Parent must correspond to an equally or more specific fact about the Child" (Ruppenhofer *et al.*, 2010). It concerns frame roles, semantic restrictions on the roles, and relations to other frames. In logical terms, $f_1$ *inherits* $f_2$ corresponds to the following FOL axiom:

$$\forall s(f_1(s) \rightarrow f_2(s)).$$

The corresponding abductive axiom is

$\textsc{Child}(e_1,x_1,\ldots,x_n) \rightarrow \textsc{Parent}(e_1,y_1,\ldots,y_m)$,
where all $x_i$ $(1 \leq i \leq n)$ and $y_j$ $(1 \leq j \leq m)$ occupy argument positions corresponding to particular frame roles so that if the role $FR_k$ $(1 \leq k \leq n)$ in the frame $\textsc{Child}$ is mapped to the role $FR_l$ $(1 \leq l \leq m)$ in $\textsc{Parent}$ then $x_k$ is equal to $y_l$. This role mapping condition is true for all the following frame relations.

*Perspective*

The use of the perspective relation "indicates the presence of at least two different points-of-view that can be taken on the Neutral frame" (Ruppenhofer *et al.*, 2010). For example, COMMERCE_BUY and COMMERCE_SELL are two perspectives of the COMMERCE_GOODS_TRANSFER frame. This relation is useful for reasoning, since some paraphrases can be analyzed in terms of perspectives. For example, *John sold a book to Mary* describes the same situation as *Mary bought a book from John*. Perspectives refer to the same situation. Therefore $f_1$ *is a perspective of* $f_2$ represents the following FOL axiom:

$$\forall s(f_1(s) \leftrightarrow f_2(s)).$$ [11]

The corresponding abductive axioms are

$$\text{NEUTRAL\_FRAME}(e_1, x_1, \ldots, x_n) \rightarrow \text{PERSPECTIVE\_FRAME}(e_1, y_1, \ldots, y_m)$$

$$\text{PERSPECTIVE\_FRAME}(e_1, y_1, \ldots, y_m) \rightarrow \text{NEUTRAL\_FRAME}(e_1, x_1, \ldots, x_n).$$

*Precedence*

The precedence relation characterizes the temporal order defined on sequences of situations. For example, FALL_ASLEEP precedes BEING_AWAKE in the SLEEP_WAKE_CYCLE scenario. In most cases where this relation is used in FN, the existence of the later situation presupposes that the preceding situation has taken place. However, the opposite also occurs, for example, GET_A_JOB presupposes BEING_EMPLOYED.

The axioms representing relations $f_1$ *precedes* $f_2$ have the following form:

(1) a situation presupposes a preceding situation
$\forall s_2(f_2(s_2) \rightarrow \exists s_1(f_1(s_1) \wedge \textit{temporally\_precedes}(s_1, s_2)))$

(2) a situation presupposes a succeeding situation
$\forall s_1(f_1(s_1) \rightarrow \exists s_2(f_2(s_2) \wedge \textit{temporally\_precedes}(s_1, s_2)))$

The predicate *temporally_precedes* can be further linked to a theory of time. The corresponding abductive axioms have the following from.

$$\text{PRECEDING\_FRAME}(e_1, x_1, \ldots, x_n) \rightarrow \text{SUCCEEDING\_FRAME}(e_2, y_1, \ldots, y_m)$$

$$\text{SUCCEEDING\_FRAME}(e_2, y_1, \ldots, y_m) \rightarrow \text{PRECEDING\_FRAME}(e_1, x_1, \ldots, x_n)$$

*Subframe*

Some frames in FN are complex and refer to sequences of situations, which can themselves be separately described as frames. The complex frame is connected to its components via

[11] This axiom does not grasp the asymmetric character of the relation between a perspectivized and a neutral frame.

the subframe relation. For example, COMMITTING_CRIME, CRIMINAL_INVESTIGATION, and CRIMINAL_PROCESS are subframes of CRIME_SCENARIO. The subframe relation embodies axioms involving the ontological *parthood* relation between situations.

For parthood, the direction of entailment is undefined. For some frame pairs the whole obviously presupposes the existence of a part. For example, GETTING_SCENARIO presupposes PRE_GETTING (possession phase), but not vice versa. $X$ getting $Z$ from $Y$ presupposes $Y$ having $Z$. Obviously, if $Y$ has $Z$ it does not mean that $X$ will get $Z$ from $Y$. For other pairs, the part presupposes the existence of the whole. For example, EMPLOYMENT_END presupposes EMPLOYMENT_SCENARIO, but not vice versa. If one stops to be employed at $X$ (he/she quites or is fired) then naturally one has been employed at $X$. On the other hand, being employed does not necessarily presupposes quitting.

Given $f_1$ is a part frame and $f_2$ is a whole frame, two axiomatizations are possible for this relation:

(1) the whole presupposes the existence of the part
$\forall s_2(f_2(s_2) \rightarrow \exists s_1(f_1(s_1) \wedge parthood(s_1,s_2)))$
(2) the part presupposes the existence of the whole
$\forall s_1(f_1(s_1) \rightarrow \exists s_2(f_2(s_2) \wedge parthood(s_1,s_2)))$

The parthood predicate can be linked to a spatial theory. The corresponding abductive axioms have the following from.

$$\text{WHOLE\_FRAME}(e_1,x_1,\ldots,x_n) \rightarrow \text{PART\_FRAME}(e_2,y_1,\ldots,y_m)$$

$$\text{PART\_FRAME}(e_2,y_1,\ldots,y_m) \rightarrow \text{WHOLE\_FRAME}(e_1,x_1,\ldots,x_n)$$

*Causation*

The relations "causative_of" and "inchoative_of" in FrameNet capture causative relationships between situation. For example, the GIVING and the GETTING frames are connected via the causative_of relation. The effect situation is always presupposed by its cause. Thus, $f_1$ *is causative of* $f_2$ and $f_1$ *is inchoative of* $f_2$ correspond to axioms of a unique form:

$$\forall s_1(f_1(s_1) \rightarrow \exists s_2(f_2(s_2) \wedge causes(s_1,s_2))).$$

The *causes* predicate can be linked to a causation theory. The corresponding abductive axiom is

$$\text{CAUSE\_FRAME}(e_1,x_1,\ldots,x_n) \rightarrow \text{EFFECT\_FRAME}(e_2,y_1,\ldots,y_m).$$

*Using*

The "using" relation in FN represents a very general kind of link. It is mostly used in cases when "a part of the scene evoked by the Child refers to the Parent frame" (Ruppenhofer *et al.*, 2010). For example, OPERATE_VEHICLE uses MOTION. In most cases, the Child frame entails the Parent frame in FN. Thus, the axioms representing relations $f_1$ *uses* $f_2$ have the following form:

$$\forall s_1(f_1(s_1) \rightarrow \exists s_2(f_2(s_2))).$$

The corresponding abductive axiom is

$\text{USING\_FRAME}(e_1, x_1, \ldots, x_n) \rightarrow \text{USED\_FRAME}(e_2, y_1, \ldots, y_m)$.

Frame relations imply more than the axioms above, which only express constraints on the situations instantiating them. When two situations actually instantiate a specific link between frames, in addition, frame role ($FR$) mappings provide information about identical entities, which are elements of both situations. So, if frame $f_1$ is related to frame $f_2$ with a relation *rel*, then, in addition to the axioms above, we have a series of axioms of the form:

$$\forall s_1 s_2((f_1(s_1) \wedge f_2(s_2)) \rightarrow (rel(s_1, s_2) \leftrightarrow \forall x(FR_1(s_1, x) \leftrightarrow FR_2(s_2, x)))),$$

where a frame role $FR_1$ in $f_1$ is mapped to a frame role $FR_2$ in $f_2$, and where *rel* stands for the predicate associated with the frame relation, e.g., *causes* for causation relations. For the inheritance and perspective relations $rel(s_1, s_2)$ should be replaced with $true$.

For example, two frames KILLING and DEATH are related with the causation relation in FrameNet so that the frame role VICTIM in the former frame is mapped to the role PROTAGONIST in the latter frame. This mapping can be converted into the following abductive axiom:

$\forall s_1 s_2((\text{KILLING}(s_1) \wedge \text{DEATH}(s_2)) \rightarrow$
$(causes(s_1, s_2) \leftrightarrow \forall x(\text{VICTIM}(s_1, x) \leftrightarrow \text{PROTAGONIST}(s_2, x))))$.

On the one hand, this axiom guarantees that given a text fragment annotated with $f_1$, if there is a relation connecting $f_1$ to $f_2$, we can correctly annotate the corresponding role fillers in $f_2$. For example, given a sentence *An avalanche killed John* we can prove *John died*. On the other hand, given a text fragment annotated with $f_1$ and $f_2$, related in FN by some frame relation, and such that all mapped frame roles in $f_1$ and $f_2$ annotate the same linguistic referents, we can infer a relation (e.g., *causes*) between the corresponding situations. For example, given a sentence *An avalanche killed John and he died* we can infer the causation relation between the killing and death events, which will not be inferred in the case of *An avalanche killed John and Mary died*.[12]

In the abductive framework, the frame role mapping happens directly through the argument mapping. Suppose that the third argument position is reserved for VICTIM in the KILLING frame and the fourth argument position is reserved for PROTAGONIST in DEATH. Then the corresponding axiom will have the following form:

$$\textsc{Killing}(e_1, x_1, \underline{x_2}, \ldots, x_n) \rightarrow \textsc{Death}(e_1, y_1, y_2, \underline{x_2}, \ldots, y_m),$$

where the fourth argument of DEATH, which is $x_2$ is equal to the third argument of KILLING.

#### 5.2.2.3 *Axioms derived from Corpora*

In addition to converting frame relations existing in FrameNet into axioms, we automatically generated frame relations not occurring in the current version of FrameNet by using clustering techniques (cf. Ovchinnikova *et al.*, 2010). Clustering was based on frame relatedness measures which have been extensively investigated by Pennacchiotti and Wirdth (2009). We applied two of these measures, which are described below, for generating clusters of semantically related frames.[13]

(1) Overlapping frame roles

Frames sharing more than $n$ infrequent[14] frame roles (FR) are considered to be semantically related and belong to the same cluster. The best result was achieved with $n = 2$. The algorithm produced 228 clusters suggesting 1497 relations not contained in the current version of FN. 100 randomly selected clusters were investigated manually by two experts which have reported that 73 clusters contain semantically related frames with the overall agreement of 0.85.

(2) Co-occurrence of lexemes evoking frames in corpora

Frequently co-occurring frames are supposed to be semantically related. Given two frames $f_1$ and $f_2$ the measure of their co-occurrence in a corpus $C$ is estimated as the pointwise mutual information (pmi):

[12] Such inferences may produce mistakes by connecting unrelated events. However, introducing these links may be useful in practice for inferring discourse relations. In order to avoid possible inference overgeneralization, one can replace equivalence with implication in the axiom under consideration.

[13] The obtained frame clusters are available online at *http://www.cogsci.uni-osnabrueck.de/∼eovchinn/FNClusters*.

[14] A frame role is infrequent if it occurs in all FrameNet frames less frequently than average.

Table 5.4 Results of FrameNet frame clustering.

| | **FR overlap** | **Lexeme co-occ.** |
|---|---|---|
| parameter value in best run | $n = 2$ | $t = -17$ |
| number of clusters | 228 | 113 |
| found relations not contained in FrameNet | 1497 | 1149 |
| semantically related clusters (out of 100) | 73 | 65 |
| overall expert agreement | 0.85 | 0.85 |

Table 5.5 Statistics for axioms extracted from FrameNet.

| **Axiom type** | **Source** | **Number of axioms** |
|---|---|---|
| Lexeme-frame mappings | FN 1.5 | 49 100 |
| Frame relations | FN 1.5 | 1 700 |
| Frame relations | Corpus | 3 600 |

$$pmi(f_1, f_2) = \log_2 \frac{|C_{f_1,f_2}|}{|C_{f_1}||C_{f_2}|},$$

where $C_{f_i}$ is the set of contexts, in which $f_i$ occurs, and $C_{f_1,f_2}$ is the set of contexts, in which $f_1$ and $f_2$ co-occur. Sentences from a newspaper corpus (Guardian, 2 600 000 sentences) were used as contexts. A frame $f$ belongs to a cluster if for all $f_i$ in this cluster $pmi(f, f_i)$ is above a threshold $t$. The best result was achieved with $t = -17$ ($-26 < pmi < -8$). The algorithm produced 113 clusters suggesting 1149 relations not contained in the current version of FN. 100 randomly selected clusters were investigated manually by two experts, which have reported that 65 clusters contain semantically related frames with the overall agreement of 0.85.

The clustering results are summarized in Table 5.4. The number of found relations not contained in FrameNet as well as the number of semantically related clusters out of the manually checked clusters suggest that the proposed clustering technique is promising for enriching the semantic structure of FrameNet.

Statistics of the axioms extracted from FrameNet is given in Table 5.5. The number of axioms is approximated to the nearest hundred.

### 5.2.3 *Axioms derived from Proposition Store*

We used the Proposition Store[15] (Peñas and Hovy, 2010) in order to generate axioms for the interpretation of dependencies between nouns such as noun compounds and possessives.

The Proposition Store was build from a collection of 216 303 New York Times articles categorized as "World" news. Around 7 800 000 sentences from this collection were parsed using the Stanford typed dependency parser.[16] The parses were used, after collapsing some syntactic dependencies, for obtaining propositions. In this approach, *propositions* are "the tuples of words that have some determined pattern of syntactic relations among them" (Peñas and Hovy, 2010).

For example, given the sentence *Steve Walsh threw a pass to Brent Jones in the first quarter*, the following propositions are extracted:

```
nvn:[Steve_Walsh:n, throw:v, pass:n]
nvnpn:[Steve_Walsh:n, throw:v, pass:n, to:p, Brent_Jones:n]
nvnpn:[Steve_Walsh:n, throw:v, pass:n, in:p, quarter:n]
```

Each proposition is tagged with its frequency in the processed document collection and with its type, which corresponds to the type of the syntactic dependency, e.g., `nvn` for `Steve_Walsh:throw:pass`, `nvnpn` for `Steve_Walsh:throw:pass:to:Brent_Jones`.[17] Moreover, general patterns considering appositions and copula verbs detected by the dependency parser were used in order to extract `is` and `has-instance` relations, e.g., `Eric_Davis:is:cornerback`, `cornerback:has-instance:Eric_Davis`.

The Proposition Store supports queries about any proposition parts. For example, given the query "which patterns include nouns *Germany* and *people*", the following patterns can be found[18]:

```
npn:[people:n, in:p, Germany]:6433
i_has_n:[Germany, people:n]:2035
nvpn:[people:n, live:v, in:p, Germany]:288
```

We used proposition patterns in order to generate abductive axioms for the interpretation of noun compounds (*nn*) and possessives (*poss*). The problem of interpreting dependencies between nouns implying implicit predicates is discussed in Sec. 2.3.2 and 8.1.3.

[15] We would like to thank Anselmo Peñas for making Proposition Store available and extracting patterns relevant for the experiments described in this book.

[16] *http://nlp.stanford.edu/software/lex-parser.shtml*

[17] `v` stands for "verb", `n` stands for "noun", and `p` stands for "preposition".

[18] `i` stands for "instance", the number after each proposition shows its frequency

It consists in finding an appropriate predicate (or a set of predicates) explicitly expressing the dependency. For example, a morning coffee is most probably a coffee drunk in the morning, while a morning newspaper is a paper read in the morning.

Patterns from the Proposition Store can help to generate hypotheses for the interpretation of noun dependencies. For example, the last pattern above including the nouns *Germany* and *people* can be converted into the following abductive axioms applicable for the interpretation of the possessive *Germany's people*:

$$people(e_2,x_2) \land live(e_3,x_2,u_1,u_2) \land in(e_4,e_3,x_1) \land Germany(e_1,x_1) \rightarrow$$
$$Germany(e_1,x_1) \land people(e_2,x_2) \land poss(e_5,x_1,x_2)$$

The abductive axioms were generated using the proposition types `nvn`, `npn`, `nvpn`, `n_has_n`, `i_has_n`, `n_has-instance_i` as follows[19]:

- `nvn` (e.g., *court accepts appeal → appeal court*):
  $noun_1(e_1,x_1) \land verb(e_3,x_1,x_2,x_3) \land noun_2(e_2,x_2) \rightarrow$
  $noun_1(e_1,x_1) \land noun_2(e_2,x_2) \land rel(e_4,x_1,x_2)$
- `npn` (e.g., *system for aircraft → aircraft system*):
  $noun_1(e_1,x_1) \land prep(e_3,x_1,x_2) \land noun_2(e_2,x_2) \rightarrow$
  $noun_1(e_1,x_1) \land noun_2(e_2,x_2) \land rel(e_4,x_1,x_2)$
- `nvpn` (e.g., *bomb explode in attack → attack bomb*):
  $noun_1(e_1,x_1) \land verb(e_3,x_1,x_3,x_4) \land prep(e_4,e_3,x_2) \land noun_2(e_2,x_2) \rightarrow$
  $noun_1(e_1,x_1) \land noun_2(e_2,x_2) \land rel(e_4,x_1,x_2)$
- `n_has_n` or `i_has_n` (e.g., *company has car → car company*):
  $noun_1(e_1,x_1) \land have(e_3,x_1,x_2) \land noun_2(e_2,x_2) \rightarrow$
  $noun_1(e_1,x_1) \land noun_2(e_2,x_2) \land rel(e_4,x_1,x_2)$
- `n_has-instance_i` (e.g., *Paolo-Maldini is captain → captain Paolo-Maldini*):
  $noun_1(e_1,x_1) \land be(e_3,x_1,x_2) \land noun_2(e_2,x_2) \rightarrow$
  $noun_1(e_1,x_1) \land noun_2(e_2,x_2) \land rel(e_4,x_1,x_2)$

For estimating the axiom weights, we used the proposition frequencies as shown in Eq. 5.1. In addition to the axioms for the interpretation of noun dependencies, axioms assuming noun dependencies were added to the knowledge base, for example:

$$Germany(e_1,x_1) \land people(e_2,x_2) \land poss(e_3,x_1,x_2) \rightarrow$$
$$people(e_2,x_2) \land live(e_4,x_2,u_1,u_2) \land in(e_5,e_4,x_1) \land Germany(e_1,x_1)$$

[19]The predicate *rel* stands for the *nn* or *poss* predicates. The predicate names $noun_1$, *verb*, and $noun_2$ should be replaced with the corresponding nouns and verbs.

The weights of these "reverse" axioms are equal to 1, because they imply assumptions of a less specific proposition. Furthermore, two general axioms assuming equivalence of the "of" prepositional phrases and possessive constructions with the weights equal to 1.2 were added to the knowledge base:

$$of(e,x_1,x_2) \rightarrow poss(e,x_1,x_2)$$
$$poss(e,x_1,x_2) \rightarrow of(e,x_1,x_2)$$

## 5.3 Ontology

As already mentioned in the earlier chapters of this book, ontologies included in the proposed integrative knowledge base are intended to represent domain-specific knowledge and are applied for the interpretation of domain-specific texts. Ontological knowledge is stored and queried separately from lexical-semantic knowledge, because the former requires a syntactically richer representation language and supports more complicated queries than the latter (cf. Sec. 3.2).

Any domain-specific ontology can be added to the proposed KB. For the experimental purposes, in this study, we used the toy OWL[20] ontology about cars, which contains descriptions of different car types and models.[21] Here is an example.

$$\textsc{Fiat\_Punto\_Grande} \sqsubseteq \textsc{Fiat} \sqcap \leq 3 \textsc{ has\_Doors} \sqcap \geq 3 \textsc{ has\_Doors}$$
$$\textsc{Fiat} \sqsubseteq \textsc{Car},$$

which means that a Fiat Punto Grande is a Fiat (which is a car) having exactly 3 doors.[22] This ontology contains 246 classes, 15 properties, and 26 instances.

In most cases, atomic primitives in an ontology are concept labels rather than words (in contrast to lexical-semantic resources), e.g., HAS_DOORS. Therefore, a mapping from lexemes to concept labels is required for ontologies to be applicable to natural language processing. Most of the mapping models imply "lexeme to concept" mappings, e.g., *car* – CAR (see Cimiano *et al.*, 2007). However, it is not always possible to find one single lexeme, which corresponds to a complex concept like HAS_DOORS or HAS_APPEARANCE. On the other hand, a lexeme may imply a complex logical formula rather than a single concept, e.g., *colored* should probably be mapped to the formula $\forall$ HAS_COLOR.($\neg$ (WHITE $\sqcup$ BLACK

[20] OWL is an ontology representation language based on Description Logics, see Sec. 4.4.
[21] *http://sisinflab.poliba.it/colucci/files/webstart/Ontology_file/car.owl*
[22] For Description Logic syntax, see Sec. 4.4.

$\sqcup$ GREY)) in the “car” context. A few approaches to ontology-based NLP presuppose a more structured mapping; for example, see (Andreasen and Nilsson, 2004).

In the proposed framework, logical forms can be mapped to ontological formulas, which enables us to map linguistic phrases to complex concept descriptions, as shown by the following examples.

(1) $\text{FIAT\_PUNTO\_GRANDE}(x_1) \rightsquigarrow fiat(e_1,x_1) \wedge punto\text{-}grande(e_2,x_2) \wedge nn(e_3,x_1,x_2)$

(2) $\text{CAR}(x_1) \rightsquigarrow car(e_1,x_1)$

(3) $(\leq 3\ \text{HAS\_DOORS} \sqcap \geq 3\ \text{HAS\_DOORS})(x_1) \wedge \text{HAS\_DOORS}\ (x_1,x_2) \rightsquigarrow$
$with(e_1,x_1,x_2) \wedge 3(e_2,x_2) \wedge door(e_3,x_2)$

Note that the mappings above are not logical axioms used for reasoning.[23] Such mappings are applied in order to enrich logical forms with ontological concept descriptions (e.g., $\leq 3$ HAS_DOORS $\sqcap \geq 3$ HAS_DOORS). However, for the abductive reasoner, these concept descriptions are just strings of symbols. They are used as formulas only when a Description Logic reasoner is employed (see Sec. 7.3). Therefore, there is no clash of two different representation formats (flat conjunctions of $n$-ary propositions and Description Logic formulas) as it might seem.

Recall that Description Logic concept descriptions are one-argument predications and Description Logic relations always take two arguments (see Sec. 4.4). Correspondingly, concept descriptions in the lexicon-ontology mapping are assigned one argument, e.g., $x_1$ referring to an entity having 3 doors is assigned to the concept description $\leq 3$ HAS_DOORS $\sqcap \geq 3$ HAS_DOORS. Ontological relations are assigned two arguments, e.g., $x_1$ referring to an entity having a door and $x_2$ referring to a door is assigned to the relation HAS_DOORS.

For the sake of experiment described in Sec. 8.3.4, we semi-automatically constructed such mappings for the mentioned ontology of the car domain. Ontological labels were translated into logical forms. First, the symbol “_” was replaced with the space symbol, e.g., FIAT_PUNTO_GRANDE was converted into *fiat punto grande*. Furthermore, concept names containing capital letters in the middle were split, e.g., FIATPUNTOGRANDE was converted into *fiat punto grande*. The resulting phrases were parsed using the *ESG* parser outputting logical forms (see Sec. 4.1.1). The corresponding logical forms were corrected manually.

[23]Therefore the symbol $\rightsquigarrow$ instead of the implication symbol $\rightarrow$ is used here.

## 5.4 Similarity Space

Hand-crafted lexical-semantic dictionaries such as WordNet and FrameNet provide both an extensive lexical coverage and a high-value semantic labeling. However, manually created resources are in principle static; updating them with new information is a slow and time-consuming process. By contrast, common sense knowledge and the lexicon are dynamic entities, which are subject to continuous changes. New words appear and existing words acquire new senses and semantic relations. This especially concerns proper names and neologisms, which constantly emerge in our shared knowledge background. Hand-crafted electronic dictionaries usually contain a very limited number of well-established and culturally important proper names, e.g., *Shakespeare*, *White House*, *Beatles*, *Lincoln* are included into WordNet. However, proper names and related associations constitute an important part of our word knowledge. For example, the name *Barack Obama* was unknown just a few years ago. Nowadays this name has strong associations shared by most of the people in our linguistic and cultural community, such as *president*, *United States*, *African American*, etc.

These newly appearing associations can be easily mined from up-to-date corpora, such as the World Wide Web corpus, by using the methods of distributional semantics (see Sec. 2.2.1). Distributional semantics allows us to define semantic similarity between two terms, which is based on the frequency of co-occurrence of these two terms in a corpus.

We propose to use semantic similarity in order to detect relationships between those parts of the discourse, which cannot be linked with the help of the axioms from the KB. In order to support reasoning, we use a similarity vector space, which can be queried for computing a similarity value between two terms or between two combinations of terms (cf. Sec. 7). There are no restrictions on methods or training corpora used to calculate the vector space. For example, for the interpretation of domain-specific texts, a domain-specific corpus can be used for computing similarities.

We used a semantic space generated with the help of Latent Semantic Analysis (Landauer, 2007). Latent Semantic Analysis (LSA) employs the vector space model from information retrieval (Salton and McGill, 1986). In its original form (Deerwester *et al.*, 1990), LSA is based on a co-occurrence matrix of terms in documents. Here, a given corpus of text is first transformed into a term$\times$document matrix, displaying the occurrences of each word in each document. The size of the matrix is determined by the number of terms in the vocabulary and the number of documents. Such a matrix is normally extremely sparse, and it is obvious that this matrix grows with the size of the training corpus. Moreover, the notion of document varies strongly over different corpora: a document can be only a paragraph, an article, a chapter, or a whole book; no hard criteria can be defined. Therefore, another type of matrix can be used, as described by Schütze (1998), Cederberg and Widdows (2003), and Wandmacher (2005), which is not based on occurrences of terms in documents but on co-occurring terms (term×term-matrix) in defined contexts. The size of the matrix is then independent of the size of the training data, so that much larger corpora can be used for training. For example, given a corpus consisting of one sentence *John read a book and the book was interesting* and the size of the context window equal to 3, the following matrix can be constructed.

| | John | read | a | book | and | the | is | interesting |
|---|---|---|---|---|---|---|---|---|
| John | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| read | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| a | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| book | 0 | 1 | 1 | 2 | 1 | 1 | 1 | 1 |
| and | 0 | 0 | 1 | 2 | 1 | 1 | 0 | 0 |
| the | 0 | 0 | 0 | 2 | 1 | 1 | 1 | 0 |
| is | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| interesting | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |

The decisive step in the LSA process is then a *singular value decomposition* (SVD) of the matrix, which enables the mapping of this matrix to a subspace. This procedure is similar to principal component analysis, cf. Berry *et al.* (1999). SVD filters out noisy elements and strengthen important connections. Furthermore, it reduces the matrix size from several thousand to a few hundred dimensions, making vector comparisons much quicker. The resulting lower-dimensional matrix is the best least-squares approximation of the original matrix (see Berry *et al.*, 1999, for formal definition of SVD). According to the proponents of LSA, SVD reduction plays an important role for the uncovering of important relations, which are hidden (or "latent") in the original matrix.

After applying SVD, each word is represented as a vector of $k$ dimensions. A difficult aspect here concerns determining the optimal number of dimensions $k$, to which the matrix will be reduced. If the reduction is too high, important relatedness information gets lost. If the reduction is too low, noisy relations are not sufficiently filtered. In most works, the parameter $k$ is heuristically set to 100–400 dimensions (cf. Deerwester *et al.*, 1990; Landauer *et al.*, 1998; Wandmacher, 2005).

Each term in the reduced matrix is represented by a vector. For every word pair $w_i, w_j$ of the vocabulary, the distance between the term vectors $\overrightarrow{w_i}, \overrightarrow{w_j}$ is used to calculate a similarity value $sim(w_i, w_j)$. Most work on LSA employ the cosine measure (Deerwester *et al.*, 1990; Landauer and Dumais, 1997; Landauer *et al.*, 1998; Wandmacher, 2005). The cosine of the angle between any two vectors $\overrightarrow{w_i}$ and $\overrightarrow{w_j}$ of dimensionality $n$ with components $w_{ik}$ and $w_{jk}$ ($k \leq n$) is defined as follows:

$$\cos(\overrightarrow{w_i}, \overrightarrow{w_j}) = \frac{\sum_{k=1}^{n} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{n} w_{ik}^2 \sum_{k=1}^{n} w_{jk}^2}}. \tag{5.2}$$

The comparison of co-occurrence vectors of terms is also referred to as *second order co-occurrence* (cf. Grefenstette, 1994), since it establishes the similarity not because two terms co-occur, but because they occur in similar environments. This is an important point, because two terms can happen to show a very high second-order similarity, even though they do not co-occur. Figure 5.2 summarizes the processing steps involved in training an LSA-based semantic space.[24]

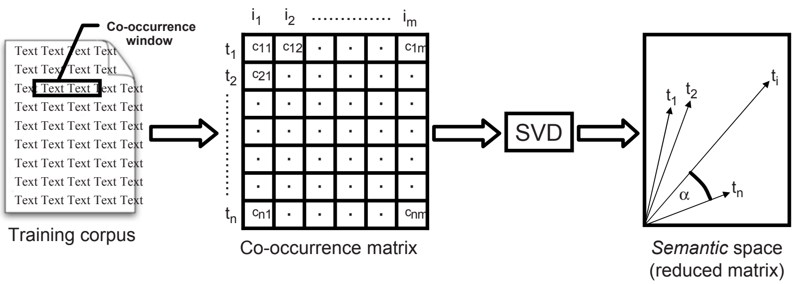


Fig. 5.2 Schematic overview on the generation of an LSA-based semantic space.

In the experiments described in Chap. 8 of this book, we used a similarity space generated using a corpus of 108M words from two British newspapers (*The Times*, *The Guardian*) of the years 1996 to 1998. A term×term co-occurrence matrix of size 80 000×3 000[25] has been generated using the *Infomap* NLP toolkit[26], developed at Stanford University's CSLI; closed-class words not occurring in the test data were disre-

[24]The scheme is provided by Wandmacher (2008).
[25]The two sets of terms need not be identical, one can also define a (usually smaller) set of index terms.
[26]*http://infomap-nlp.sourceforge.net/*

garded.[27] The vocabulary as well as the index terms were determined via frequency, terms occurring less than 24 times in the corpus were excluded from the vocabulary. The similarity space was calculated for the co-occurrence window of 75 words, the window did not cross article boundaries.[28] The matrix was reduced by SVD to 300 dimensions; term similarity was determined by measuring the cosine of the angle between the corresponding term vectors. Since negative cosine values can occur but are meaningless for similarity measurements (i. e. terms having a negative similarity value are not more dissimilar than those having a value of 0), negative values are set to 0.

## 5.5 Concluding Remarks

In this chapter, we described the proposed integrative knowledge base. The KB contains abductive axioms derived from such handcrafted lexical-semantic resources as WordNet (Sec. 5.2.1) and FrameNet (Sec. 5.2.2). WordNet's strong part is its large lexical and conceptual coverage reflected in the detailed hierarchical structure, which is especially elaborate for nouns. Three types of axioms were derived from WordNet: 1) lexeme-synset mappings, 2) relations defined on synsets and word senses, and 3) synset definitions represented by logical forms. In contrast to WordNet, FrameNet does not operates with atomic concepts (like synsets), but with structured conceptual entities which are frames. The lexical coverage of FrameNet yields to the coverage of WordNet, but its structural richness makes FrameNet especially valuable for reasoning. The following types of axioms were derived from FrameNet: 1) lexeme-frame mappings and 2) frame relations.

The set of abductive axioms in the KB was extended with the axioms for interpretation of noun compounds and possessives generated using the Proposition Store (Sec. 5.2.3). These axioms were applied in an experiment on noun dependency interpretation described in Sec. 8.3.3.

A separate module of the integrative knowledge base contains a domain-specific ontology used in an experiment on interpretation of a domain-specific text (see Sec. 8.3.4). This module also contains the lexicon-ontology mapping, which maps logical forms to ontological concept descriptions. Moreover, the KB includes a similarity matrix used to infer discourse dependencies, which cannot be recovered on the basis of the axioms.

[27] We would like to thank Tonio Wandmacher for making the co-occurrence matrix available.

[28] Although there have been attempts on investigating how the difference in the context window size influences the resulting similarity space (see, for example, Bullinaria and Levy, 2007), this issue is still far from being clarified. In the described experimental settings, we use the 75 window, because it proved to give the best results in the task of predicting human associations where the same training corpus was used, see Wandmacher *et al.* (2008).

The resulting integrative knowledge base has three separate modules requiring different representation formats. The lexical-semantic module contains abductive axioms, i.e. implications which contain flat conjunctions of *n*-ary predications on the left and right hand sides, such that weights are assigned to the predications on the left hand side. The ontological module is represented using an OWL language based on Description Logics. The distributional module contains vectors in a similarity space.

## Chapter 6

# Ensuring Consistency

Ensuring consistency of a knowledge base is crucial for reasoning. Knowledge axiomatization mistakes may block intended inferences and generate unintended ones. Since knowledge bases appropriate for natural language understanding should be large enough to support sufficient lexical and conceptual coverage, their manual verification and repair is a challenging and time-consuming task. Therefore automatic procedures for detecting and repairing inconsistencies are preferred over manual ones. However, as the reader will see in the following sections, fully automatic diagnosis and correction of a KB is not always possible.

The notion of *inconsistency* implies multiple axioms in the KB, which contradict each other. Contradictions can occur on different levels; correspondingly, the following types of inconsistency can be distinguished[1]:

- *Structural inconsistency* is defined with respect to the underlying representation language. A knowledge base is structurally inconsistent, if it contains axioms violating the syntactic rules of the representation language.
- *Logical inconsistency* is defined on the basis of formal semantics of the knowledge base. A KB is logically inconsistent if it has no model (see Sec. 6.2).
- *Conceptual inconsistency* is related to underlying ontological constraints (see Sec. 6.1).
- *User-defined* inconsistency is related to application context constraints defined by the user.

Structural inconsistency is easy to detect and to repair using a syntax checker. Detecting logical inconsistency requires employing a reasoning machine. Automatic repair of detected contradictions is far from trivial and may have multiple solutions. Conceptual inconsistency is even more challenging, because it requires deep understanding of ontolog-

[1] Haase *et al.* (2005), for example, distinguish structural, logical, and user-defined inconsistency.

ical modeling principles. User-defined inconsistency is application-specific and cannot be captured by general domain-independent mechanisms.

In this book, we focus on logical and conceptual inconsistency of a knowledge base relevant for natural language understanding. Concerning conceptual inconsistency, we consider conceptual errors in relations going beyond taxonomy, which have not been discussed in the previous literature, and propose new ontological constraints enabling detection of these errors. This study is carried out with a particular focus on NLP applications, i.e. the proposed ontological principles are aimed at enabling inferences relevant for natural language understanding. As for logical inconsistency, we develop a new automatic procedure for repairing logically inconsistent ontologies. The procedure eliminates logical contradictions and is maximally knowledge-preserving.

## 6.1 Conceptual Inconsistency of Frame Relations

As already mentioned in the introductory chapter of this book, conceptual consistency is related to formal ontological foundations of modeling the world. Since there are yet no generally accepted standards of knowledge engineering, engineers cannot avoid being subjective when axiomatizing concepts and concept definitions. Without any generally accepted theory of knowledge modeling, it is easy to make conceptual mistakes, which will result in misleading inferences.

A few researchers working in the area of the overlap of computer science and philosophy have been focusing on formulating ontological principles for designing taxonomies. The most well-known methodology for "cleaning up" taxonomies, called OntoClean, was developed by Guarino and Welty (2004). The basis of OntoClean are the domain-independent meta-properties of concepts: *identity*, *unity*, *rigidity*, and *dependence*. These meta-properties are used to define taxonomic constraints.

For example, a property is rigid if it must be true of all its possible instances in all possible words, e.g., the property *being a human* is typically rigid, since every human is necessarily so. Properties that are not essential to all their instances are called anti-rigid. *Being a student* is anti-rigid, because every instance of *student* can cease to be a student in a suitable state of affairs. According to OntoClean, an anti-rigid property cannot subsume a rigid property. Thus, the property *being a student* cannot subsume the property *being a human*. No instance of *human* can cease to be a human. If all humans were necessarily students, then no person could cease to be a student, which is inconsistent.

Assigning meta-properties is far from being trivial; it requires a deep understanding of the ontological nature of every concept. There are only a few attempts to evaluate conceptual consistency of an ontology automatically. Völker *et al.* (2008) employ lexico-syntactic patterns for capturing the meta-properties of ontological concepts. For example, the following intuition is used: If any individual can become or stop being a member of a certain class, then it holds that the membership of this class is not essential for all its individuals. Based on this intuition, the following patterns are used to obtain negative evidence with respect to rigidity: `is no longer (a|an)? CONCEPT`, `became (a|an)? CONCEPT`, `while being (a|an)? CONCEPT`.

Verdezoto and Vieu (2011) propose to contrast several sources of knowledge and automatically check their coherence in order to spot errors in the taxonomic structure. In particular, this work focuses on the inheritance and meronymy relations in WordNet and in automatically generated resources. One of the proposed ontological constraints states that a physical entity cannot be a part of an abstract entity and vice versa. If this constraint is violated, then either the taxonomic or the meronymic link is wrong. For example, the synset containing *balkan_wars* (an abstract entity) is a meronym of the synset containing *balkan_peninsula* (a physical entity) in WordNet. Verdezoto and Vieu (2011) claim that in this case meronymy is confused with the relation “is located in”.

All previous research on conceptual consistency concentrates on atomic concepts (e.g., *student*, *human*, *balkan_wars*) and, in overwhelming majority of cases, on the inheritance relation only. We want to make a further step in the direction of formulating constraints on semantic relations and consider more structured conceptual entities, which are frames. Besides the inheritance relation, we focus on a wider range of semantic relations, e.g., causation, temporal precedence, perspective. It is important to note that all conceptual problems occurring in the taxonomy of atomic concepts are also relevant for the relational network of frames, which introduces additional challenges due to its higher structural complexity.

A relational network of frames is considered as represented in FrameNet (Ruppenhofer *et al.*, 2010), which generalizes over predicate-argument constructions in English. We categorize conceptual problems occurring in FrameNet (FN) and perform a case study on “cleaning up” the FN relational network.[2]

The proposed methodology is based on an ontological analysis, which presupposes a) studying frames on the basis of formal principles and ontological relations like dependence, parthood, participation, etc. and b) possibly linking frames with categories in a

[2]This work was done in collaboration with the Laboratory of Applied Ontology (*http://www.loa-cnr.it/*) and first published in Ovchinnikova *et al.* (2010).

formal ontology. The benefits of using ontological principles for constraining computational lexical resources have been discussed in the literature, see, for example, Prevot *et al.* (2009). For our purposes we used the Descriptive Ontology for Linguistic and Cognitive Engineering, DOLCE (Masolo *et al.*, 2003), the ontology, which has been successfully applied for achieving a formal specification of WordNet (Oltramari *et al.*, 2002).

### 6.1.1 *Ontological Status of Frames*

In order to define constraints on frame relations, one needs first to understand what the frames describe, i.e. what the frame relations actually relate. FrameNet frames abstract from a special kind of natural language expressions, namely from predicates with their arguments in the sense of linguistic semantics.[3] Natural language expressions refer to situations in a world. The term *situation* is borrowed from Situation Theory:

> The world consists not just of objects, or of objects, properties and relations, but of objects having properties and standing in relations to one another. And there are parts of the world, clearly recognized (although not precisely individuated) in common sense and human language. These parts of the world are called situations. (Barwise and Perry, 1980)

Thus, frames abstracting from natural language expressions describe types of situations. Therefore we analyze and decompose frames in terms of situations and their parts, which can be described by predicate-argument constructions in natural language. In order to characterize situations corresponding to frames we employ categories elaborated in the framework of DOLCE, the ontology, which has been designed for capturing the ontological categories underlying natural language and human common sense.

The most important DOLCE categories, which we refer to are *perdurant*, *endurant*, *quality*, and *abstract*.[4] The distinction between perdurants and endurants, or, in other words, between events and objects, is related to their behavior in time. At any time an *endurant* is present, all parts of the endurant are present too. *Perdurants* are only partially present at every time moment, i.e. some of their temporal parts (their previous or future phases) may be not present. For example, if somebody reads a book then the book is wholly present at a given time during reading, while some temporal parts of the reading are not. The main relation between endurants and perdurants is that of *participation*: an endurant exists in time by participating in some perdurant. *Qualities* are entities, which can be perceived or measured: shapes, sounds, weights, etc. They refer to features of specific entities

[3]In FrameNet, single predicate-argument constructions are considered as represented by phrases and clauses. There is no frame abstracting from a complex sentence like *John went to the bank and Bill stayed at home*, but there are frames abstracting from noun phrases like *the young man*.

[4]For more details see (Masolo *et al.*, 2003).

and exist as long as these entities exist. *Abstract* entities in DOLCE are entities that neither have spatial nor temporal qualities and are not qualities themselves (e.g., facts, sets). Figure 6.1 shows the DOLCE basic categories.

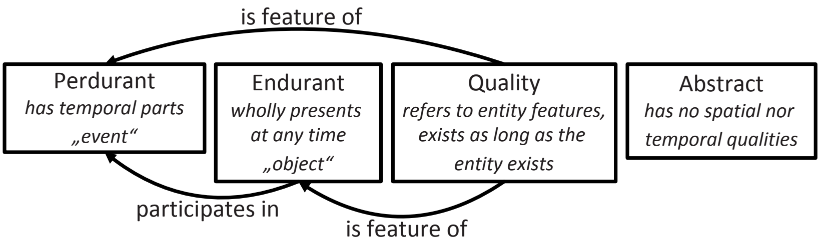


Fig. 6.1 DOLCE basic categories.

Barwise and Perry's situations are perdurant-like entities, so it seems natural to consider frames as denoting types of perdurants. As in all theories of events, perdurants in DOLCE happen in time, and this feature is clearly present in many frames. However, for some FrameNet frames this is less clear. For instance, some frames are triggered by nouns denoting the category of an endurant, e.g., PEOPLE (*a man*), and it is rather debatable whether a situation like *John being a man* is located in time.

Other frames describe relations between perdurants, e.g., RELATIVE_TIME (my birthday preceded Lea's arrival), which the large majority of authors would be reluctant to consider as some sort of "higher-order" perdurant, as explained in (Pianesi and Varzi, 2000). In fact, in FrameNet, such frames are never involved in temporal frame relations like precedence or causation, which instead freely apply to frames clearly describing perdurants. Thus, it is safer to assume that "situations" used in FN denote types of perdurants or facts, the latter being a category of abstract entities, present in DOLCE but not yet really axiomatized.[5]

Some frames seem to denote only perdurant types, others only fact types, and yet others would seem to group together situations of both sorts. For example, RECIPROCALITY is a very general frame describing a symmetrical relationship. It can refer to an event like *chatting* and to a relation like *being similar*.

Based on the categories described above we define the following main types of situations, which can be used for classifying most of the FN frames:

[5]Future work on facts is clearly needed, and such work is expected to enrich the present analysis of situations in FrameNet.

(1) **“Event”** situation: a perdurant with its participants, e.g., the SELF_MOTION frame (*John runs in the park*)
(2) **“Object”** situation: a fact that an endurant has some non-temporal property, for instance the property of being of a given category, e.g., PEOPLE (*a man*)
(3) **“Quality”** situation: a perdurant or a fact involving the attribution of a temporal or non-temporal quality to an endurant or a perdurant, e.g., COLOR (*This rose is red*, perdurant), DURATION_RELATION (*The war lasted four years*, fact)
(4) **“Relation”** situation: a perdurant or a fact involving a relation between endurants and/or perdurants, e.g., PART_WHOLE (*This park is a part of the town*, fact), LOCATIVE_RELATION (*Lea is next to John*, perdurant), RELATIVE_TIME (*my birthday preceded Lea’s arrival*, fact)

### 6.1.2 *Constraints on Frame Relations*

In the following, we define constraints on semantic relations between frames, which are based on the situation types listed above and on the axiomatization of frame relations proposed in Sec. 5.2.2.

*Inheritance of frames*

Frame relations implying identity of situations to, which both related frames refer (e.g., inheritance or perspective) presuppose that the related frames refer to situations of the same type. Violation of this constraint may imply wrong conceptualization.

For example, the frame DURATION_RELATION in FrameNet (evoked by the verbs *last* and *persist*, e.g., *The war lasted four years*) cannot inherit from the frame EVENT (evoked by *occur*, *happen*, *take place*; e.g., *The war took place*), because the former frames describes situations of the type “Quality” and the latter describes “Events”. Thus, both frames focus on different aspects. The frame DURATION_RELATION focuses on temporal qualities of a perdurant and the frame EVENT refers to a perdurant itself.

However, there might be cases when a cross-type inheritance is justified. For example, the frame RECIPROCALITY is a very general non-lexicalized frame, which FrameNet defines as characterizing “states-of-affairs with protagonists in relations with each other that may be viewed symmetrically”. This frame subsumes both frames of the type “Relation”, e.g., SIMILARITY evoked by such lexemes as *alike*, *differ*, *difference*, and frames of the type “Event”, e.g., CHATTING evoked by *chat*, *talk*, *speak*. The core feature of RECIPROCALITY is the exchangeability of its core role (PROTAGONIST_1 and PROTAGONIST_2) fillers. This feature is relevant both for the frame SIMILARITY and for the frame CHATTING. Thus, *X is similar to Y* implies *Y is similar to X* as well as *X chats with Y* implies *Y chats with X*.

*Temporal relations*

Frame relations presupposing a temporal consequence of the related situations (e.g., causation or temporal precedence) can be defined only on frames referring to perdurants, because only perdurants exist in time. For example, let us consider the semantic relation between the frames BIRTH referring to situations of the type “Event” and PEOPLE referring to “Objects”. A precedence link, which might seem to be possible here is rejected because “Object” frames cannot be involved in temporal relations. An intuitive explanation is that birth can precede people’s lives but not people themselves. Instead a dependency link could be chosen for the BIRTH-PEOPLE relation.

*Compliance with axiomatization*

In Sec. 5.2.2, an axiomatization of frame relations is proposed. Naturally, if two frames $f_1$ and $f_2$ are related by a relation, then the axioms should apply to all instantiations of $f_1$ and $f_2$. Let us consider an example of a possible violation of the axioms proposed in Sec. 5.2.2. Consider the GETTING_SCENARIO frame, which has three subframes in FN: PRE_GETTING (*X possesses Z*), GETTING (*Y gets Z from X*) and POST_GETTING (*Y possesses Z*). Subframes are supposed to be phases of a complex scenario. However, one could doubt that GETTING_SCENARIO is introduced correctly, because the subframe relation implies that the whole period of time when *X* or *Y* possessed *Z* is included into the getting scenario, which may not necessarily reflect the intended meaning of getting. Precedence relations may be enough here.

*Necessary vs. typical relations*

Relational network of frames should distinguish between typical and necessary relations. For example, a state “being recovered” (RECOVERY frame) is necessary preceded by “having a disease” (MEDICAL_CONDITIONS frame), while “being treated” (CURE frame) only typically causes “being recovered”. The current version of FrameNet does not support this distinction. Typical dependencies are most frequently captured by the “Using” links.

*Inheritance of frame roles*

Besides defining the correct type of relation between two frames, it is important to ensure that frame roles of related frames are linked in an appropriate way. Inferences with frames are extensively based on frame roles, since fillers of the linked roles are supposed to refer to the same entity. Therefore, if a role $FR_1$ in a frame $f_1$ is linked to $FR_2$ in $f_2$ where $f_2$ is inferred from $f_1$ then everything, which is true about $FR_2$ must be also true for $FR_1$. For checking whether a link between two frame roles is reasonable, we propose to assign ontological categories to the possible fillers of the corresponding roles and to find out whether inheritance between these categories is justified. In order to achieve this, the OntoClean methodology (Guarino and Welty, 2004) can be applied.

Some of the frame roles in FN are already typed with semantic types, which are organized in a small hierarchy of around 40 nodes. For example, the VICTIM role in the frame KILLING is typed as *Sentient*. For most of the roles the typing is missing. Moreover, the FN developers admit that the current hierarchy of semantic types is incomplete and suggest to use WordNet instead: "Because we cannot anticipate all of the Semantic Types that will be useful for tagging [frame roles], it will certainly also be desirable to categorize the fillers of our [frame roles] using WordNet" (Ruppenhofer *et al.*, 2010). An alternative to using WordNet would be using the basic categories of DOLCE as described in (Oltramari *et al.*, 2002).

Let us consider an example. In FrameNet, the PART_WHOLE frame inherits from PART_PIECE with the WHOLE role linked to SUBSTANCE. Looking at the lexemes evoking PART_PIECE and the corresponding annotated examples we conclude that the WHOLE role can be filled either by entities of the type *Amount of matter* (*piece of cake*) or *Mental Object* (*snippet of knowledge*), while WHOLE in PART_WHOLE can be filled by *Physical object* (*body part*), *Mental Object* (*part of my idea*), *Process* (*part of the interview process*). Thus, WHOLE covers a wider range of ontological categories than SUBSTANCE and therefore cannot inherit from SUBSTANCE.

The proposed "cleaning up" methodology can be summarized as follows. Given two frames $f_1$ and $f_2$ connected with a relation $r$

(1) define the types of situations that instantiate $f_1$ and $f_2$;
(2) if $r$ is a temporal relation (causation or precedence) make sure that both $f_1$ and $f_2$ refer to perdurant situations;
(3) define whether $r$ has a necessary or a typical character;

(4) check whether the axioms listed in Sec. 5.2.2 apply to all instantiations of $f_1$ and $f_2$;
(5) if two frame roles $fr_1$ belonging to $f_1$ and $fr_2$ belonging to $f_2$ are mapped by $r$, then
a) assign ontological types to fillers of $FR_1$ and $FR_2$ in the available FN-annotations,
b) check whether inheritance can be defined on these ontological types.

### 6.1.3 *Reasoning-Related Conceptual Problems in FrameNet*

For finding further conceptual problems in the current version of FrameNet, which prevent proper reasoning, we investigated the FrameNet-Annotated corpus for Textual Entailment (FATE), manually annotated with frame and role labels (Burchardt and Pennacchiotti, 2008).[6]

We manually analyzed the cases when text $T$ was known to entail hypothesis $H$ (400 pairs) aiming to find out whether the matching strategy as described in Sec. 5.2.2 is sufficient for establishing entailment.[7] In 170 cases direct matching was possible. For 131 pairs this approach does not work because of a) annotation problems, such as mismatch in role assignment or missing annotation; b) different conceptualizations of $T$ and $H$ resulting in different, semantically unrelated, framings. For 99 pairs the same facts in $T$ and $H$ were represented by different frames, which are related semantically and could be mapped on each other with the help of reasoning. FrameNet relations enable correct inferences only for 17 such pairs. In the following, we categorize the problems discovered in the remaining 82 pairs.

*Lack of axiomatization*

A complex axiomatization is needed for capturing some of the frame relations.[8] For instance, a relation can presuppose restrictions on role fillers. In the example below, COMMERCE_BUY and BEING_EMPLOYED need to be matched.

**T:** ... [*PeopleSoft*]$_{\text{BUYER}}$ [*bought*]$_{\text{COMMERCE\_BUY}}$ [*JD Edwards*]$_{\text{GOODS}}$ ...

**H:** [*JD Edwards*]$_{\text{EMPLOYEE}}$ [*belongs*]$_{\text{BEING\_EMPLOYED}}$ [*to PeopleSoft*]$_{\text{EMPLOYER}}$.

In the general case, if the role BUYER in the COMMERCE_BUY frame is filled by an organization and the role GOODS is filled by a person, then this frame can be considered to be semantically equivalent to the HIRING frame, which is related to BEING_EMPLOYED via COMMERCE_SCENARIO. Since proper axiomatization of all such relationships is highly

[6] For a detailed description of the FATE data set, see Sec. 8.1.2.1.
[7] Performing this analysis, we ignored aspects related to matching role fillers, which have different lexical and syntactic realization, e.g., *at least 11 people* and *humans*.
[8] Similar observations are summarized by Clark *et al.* (2007).

time-consuming, frame similarity measures allowing us to map typically rather than necessarily related frames can be used to resolve such cases.

Furthermore, such frame roles as PLACE, TIME, CAUSE, which are parts of many frames, often correspond to modifiers (e.g., adjectival or adverbial phrases). However, the description of temporal, spatial and other features of an event can also be the primary object of a sentence, giving rise to "attribute frames". In *H* of the example below, the location *China* is annotated with the BEING_LOCATED frame, while in *T* it fills the PLACE role of the BUSINESS frame. For resolving such cases one needs to have links between general "attribute frames" and some of the roles of specific frames.

**T:** ... [*First Automotive Works* [*Group*]$_{\text{BUSINESS}}$]$_{\text{BUSINESS\_NAME}}$, [*China's*]$_{\text{PLACE}}$ *vehicle maker...*

**H:** [*First Automotive Works Group*]$_{\text{THEME}}$ *is* [*based*]$_{\text{BEING\_LOCATED}}$ [*in China*]$_{\text{LOCATION}}$.

*Incompleteness of frame relations*

Some frames, which are currently not linked in FN suggest themselves to be mapped on each other. In the example below, the frames SURVIVING and RECOVERY are not connected in FN. Therefore it is impossible to infer that *H* is entailed in *T*.

**T:** ...[*people*]$_{\text{SURVIVOR}}$ *who* [*survive*]$_{\text{SURVIVING}}$ [*Sars*]$_{\text{DANGEROUS\_SITUATION}}$ ...

**H:** [*Those*]$_{\text{PATIENT}}$ *who* [*recovered*]$_{\text{RECOVERY}}$ [*from Sars*]$_{\text{AFFLICTION}}$ ...

This problem can be solved by adding more relations extracted automatically as proposed in Sec. 5.2.2.

*Frame relations vs. lexical relations*

One crucial problem of FrameNet is that its frame relations cannot be propagated on the lexical items assigned to frames. For example, the MOTION_DIRECTIONAL frame with such lexical elements as *fall*, *rise* and *drop* assigned to it inherits from the frame MOTION with the lexemes *move*, *go*, *fly*, *roll*, *zigzag*. Every falling, rising, or dropping can be seen as a type of moving, but definitely not as a type of rolling or zigzagging which will inevitably follow by application of the MOTION_DIRECTIONAL-MOTION relation to the textual data. Unfortunately, FrameNet does not mark "generic" lexemes being representative for the meaning of the frame, which would help to avoid this problem.

Furthermore, FrameNet sometimes assigns lexemes having quite an opposite meaning to the same frame. For example, the adjectives *ill* and *healthy* are both assigned to the frame MEDICAL_CONDITIONS, which makes the inference using its relation to the RECOVERY (recovering presupposes being ill) frame problematic.

### 6.1.4 *Case Study*

In order to demonstrate how the proposed methodology for cleaning up frame relations works in practice, we apply it to a cluster of FrameNet frames related to the concept of medical treatment with RECOVERY as the central frame. Figure 6.2 (top) shows the frames, which are related to RECOVERY in FrameNet by a path of 1 or 2 relations.

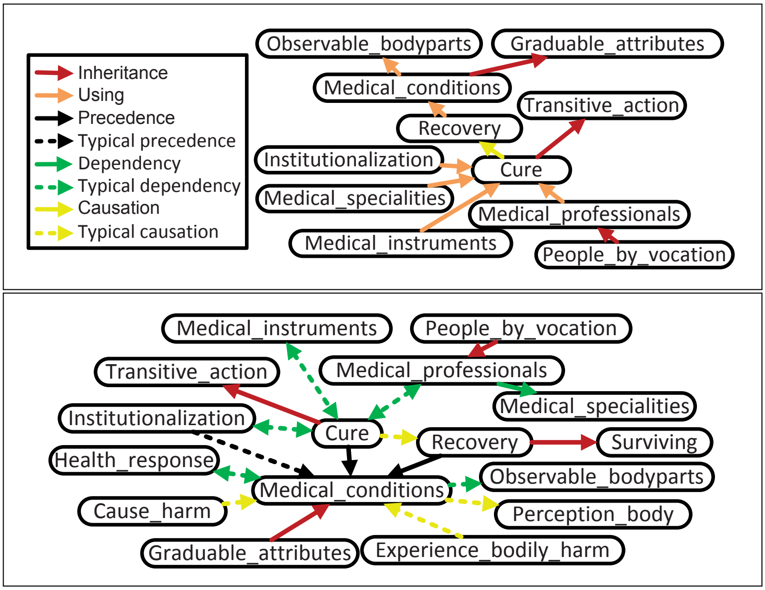


Fig. 6.2 “Medical” cluster: frame relations from FrameNet (top) enriched and cleaned up (bottom).

The clustering algorithms described in Sec. 5.2.2 have additionally discovered semantic relations between RECOVERY and the following frames: HEALTH_RESPONSE, CAUSE_HARM, EXPERIENCE_BODILY_HARM, SURVIVING, PERCEPTION_BODY.

Figure 6.2 (bottom) shows the “cleaned up” relational network in the medical cluster. More relations were added, which link the frames suggested by the clustering algorithms to the frames described above. All but one Using links were replaced with the general dependency relation. The RECOVERY-MEDICAL_CONDITIONS link was substituted with the precedence relation because of its clearly temporal character (having a disease always precedes recovery). The causation link between CURE and RECOVERY was replaced by the typical causation.

All inheritance links are left as they are, because none of them violates the proposed constraints. Let us illustrate it on the example of the CURE-TRANSITIVE_ACTION link. TRANSITIVE_ACTION and CURE are both of the type "Event" such that HEALER in CURE is linked to AGENT in TRANSITIVE_ACTION, TREATMENT is linked to CAUSE and PATIENT is linked to PATIENT. Now we need to type linked frame elements and apply the OntoClean methodology to the corresponding semantic types. Both HEALER and AGENT are of the semantic type *Sentient* in FN. TREATMENT and CAUSE frame roles are both not typed, but CAUSE is explicitly described to be an event in the TRANSITIVE_ACTION frame. The FN annotation shows that the TREATMENT frame role can be filled by both a verb phrase (referring to an event) or by a noun phrase, e.g., [*These herbs*]$_{\text{TREATMENT}}$ *can cure insomnia*. However, the annotation examples show that in the noun phrase case there is always an implicit treatment event involved in the curing situation, e.g., a treatment action performed with herbs. Thus, it seems to be reasonable to type TREATMENT with a semantic type *Event*.

The restructured "medical" cluster was evaluated in an experiment on recognizing textual entailment, which is described in Sec. 8.2.1.

## 6.2 Logical Inconsistency in Ontologies

Logical contradictions in a KB are closely related to negated axioms included in this KB. These axioms are the main source of logical inconsistencies. For example, a KB containing axioms $A$ and $\neg A$ is logically inconsistent. Logical contradictions can arise only if the underlying representation language is expressive enough to allow negation. Relations derived from lexical-semantic resources involve no negation, therefore the problem of logical inconsistency affects ontologies only.

In this study, we employ Description Logic (DL) ontologies. Therefore the proposed procedure for detecting and repairing inconsistencies is modeled in the framework of Description Logics (see Sec. 4.4). Given an inconsistent ontology, our goal is to rewrite it automatically in order to obtain a consistent one, according to the following criteria:

- The performed changes have to be relevant and intuitive.
- The changed ontology is formalized in a Description Logic language.
- As few pieces of information as possible are removed from the ontology.

In Description Logics, the notion of consistency is defined for the knowledge base containing an assertion box (ABox) and a terminological box (TBox) (see Sec. 4.4). An ABox is inconsistent only if it contains an individual instantiating an unsatisfiable concept description, for example, $\{(Person \sqcap \neg Person)(John)\}$, where $Person \sqcap \neg Person$ is obviously unsatisfiable. Most of the inconsistencies are caused by contradictions in TBoxes. For example, if an ontology contains the ABox $\mathscr{A} = \{Person(John), Dog(John)\}$ and the TBox $\mathscr{T} = \{Dog \sqsubseteq \neg Person\}$, then the ontology is inconsistent, because the individual *John* instantiates the inconsistent concept description $Person \sqcap Dog$. Thus, given a knowledge base $\mathscr{K} = (\mathscr{T}, \mathscr{A})$ such that $C(A) \in \mathscr{A}$ and $C$ is unsatisfiable, resolving inconsistency of $\mathscr{A}$ can be reduced to resolving unsatisfiability of $C$.

In practice, logical conflicts can be caused by the following reasons.

- Accidental errors
  Errors in the automatic ontology learning procedure or mistakes of the ontology engineer can generate random unintended contradictions. For example, if *Bear* is misspelled as *Beer* in the axiom $Beer \sqsubseteq Animal$, then the concept of *Beer* in the ontology can happen to be subsumed both by *Drink* and *Animal*.
- Polysemy problem
  Suppose the concept *Tree* is declared is a subconcept both of *Plant* and of *Structure* (where *Plant* and *Structure* are defined as disjoint). These two meanings of *Tree* are correct, but the resulting terminology is logically unsatisfiable.
- Generalization mistakes
  Finally, there is a set of problems related to generalization mistakes, i.e. ontological definitions, which do not take into account exceptions. This type of problems is considered below.

Suppose an ontology contains the following facts.

**Example 6.1.**

| | |
|---|---|
| $(1)\ Bird \sqsubseteq CanFly$ | (*All birds can fly.*) |
| $(2)\ CanFly \sqsubseteq CanMove$ | (*If a creature can fly then it can move.*) |
| $(3)\ Canary \sqsubseteq Bird$ | (*All canaries birds.*) |
| $(4)\ Penguin \sqsubseteq Bird \sqcap \neg CanFly$ | (*All penguins are birds and cannot fly.*) |

The statement *birds can fly* (1) in Ex. 6.1 is too general. If an exception *Penguin* (a bird that cannot fly) is added, the terminology becomes unsatisfiable.

In the past few years, a number of approaches to automatic ontology debugging have been suggested, see Bell *et al.* (2007) for a detailed overview. A technique to find a minimal set of axioms that is responsible for logical contradictions in a terminology was first proposed by Schlobach and Cornet (2003). Several other debugging methods are concerned with explanation services that are integrated into ontology developing tools (Haase *et al.*, 2005; Wang *et al.*, 2005). However, these approaches either do not provide solutions of how to fix the discovered contradictions or just propose to remove a problematic part of an axiom, although removed parts of axioms can result in a loss of information. Considering Ex. 6.1, if the concept $CanFly$ is removed from axiom 1, then the entailments $Bird \sqsubseteq CanMove$ and $Canary \sqsubseteq CanFly$ are lost.

The approaches of the second type use several well-known techniques from non-monotonic reasoning, like default sets (Heymans and Vermeir, 2002) or epistemic operators (Katz and Parsia, 2005). Unfortunately, these approaches go beyond the expressive power of Description Logics. The disadvantage is that standard DL-reasoners cannot be easily used for these extensions.

Finally, different techniques for rewriting problematic axioms were proposed (Lam *et al.*, 2006; Ovchinnikova and Kühnberger, 2006). Besides the detection of conflicting parts of axioms, a concept is constructed that replaces the problematic part of the chosen axiom. Lam *et al.* (2006) extend a tableau algorithm in order to find sets of axioms causing inconsistency and the set of “helpful” changes that can be performed to debug the terminology. This approach keeps the entailment $Bird \sqsubseteq CanMove$, but not $Canary \sqsubseteq CanFly$ in Ex. 6.1.

In this section, we summarize the approach to resolve overgeneralized concepts conflicting with exceptions as well as to treat polysemous concept names. This approach was technically introduced in (Ovchinnikova and Kühnberger, 2006) and developed in (Ovchinnikova and Kühnberger, 2007). Besides rewriting problematic axioms, a split of an overgeneralized concept $C$ into a more general concept (not conflicting with exceptions) and a more specific one (capturing the original semantics of $C$) is proposed. This allows us to preserve the entailment $Canary \sqsubseteq CanFly$.

### 6.2.1 *Resolving Logical Inconsistencies*

It is important to understand, which solution for resolving inconsistencies is appropriate from a pragmatic point of view. In general, accidental mistakes cannot be fixed automatically. The polysemy problem can be resolved by renaming concepts, which have polysemous names (e.g., by introducing *TreePlant* and *TreeStructure*). After splitting the problematic concept (e.g., *Tree*) into two concepts with different names (e.g., *TreeStructure* and *TreePlant*) it is necessary to find out which one of the definitions and subconcepts of the original concept refers to, which of the new concepts. This can be done either by the ontology engineer or with the help of additional knowledge about the usage context of this concept in external resources.

The overgeneralization problem requires a complex treatment. Let us consider a couple of examples. Concerning Ex. 6.1, it seems to be obvious that the axiom $Bird \sqsubseteq CanFly$ has to be modified, since this axiom contains overgeneralized knowledge. Simply deleting this axiom would result in the loss of the entailments $Bird \sqsubseteq CanMove$ and $Canary \sqsubseteq CanFly$, although both entailments do not contradict the axiom $Penguin \sqsubseteq \neg CanFly$. A possible solution is to replace the problematic part of the overgeneralized definition of the concept *Bird* (namely $CanFly$) with its least subsumer that does not conflict with *Penguin*. In Ex. 6.1, the concept description $CanMove$ is precisely such a subsumer. Unfortunately, the simple replacement of $CanFly$ by $CanMove$ in axiom (1) is not sufficient to preserve the entailment $Canary \sqsubseteq CanFly$. We suggest to introduce a new concept $FlyingBird$ that preserves the previous meaning of *Bird* and subsumes its former subconcepts as shown in 6.2 below:

**Example 6.2 (Rewritten Ex. 6.1).**

| | |
|---|---|
| (1) $Bird \sqsubseteq CanMove$ | *(All birds can move.)* |
| (2) $CanFly \sqsubseteq CanMove$ | *(If a creature can fly then it can move.)* |
| (3) $Canary \sqsubseteq FlyingBird$ | *(All canaries are flying birds.)* |
| (4) $Penguin \sqsubseteq Bird \sqcap \neg CanFly$ | *(All penguins are birds and cannot fly.)* |
| (5) $FlyingBird \sqsubseteq Bird \sqcap CanFly$ | *(All flying birds are birds that can fly.)* |

The situation is different for multiple overgeneralizations. Example 6.3 below shows a case where two overgeneralized definitions of the same concept conflict with each other. Axioms (1) and (6) contain overgeneralized definitions; they both describe a special case rather than a general rule. The subsumption $Child \sqsubseteq \forall likes.(Icecream \sqcap \neg Icecream)$ follows from the inconsistent terminology.

**Example 6.3.**

(1) $Child \sqsubseteq \forall likes.Icecream$ (*Children like only icecream.*)

(2) $Icecream \sqsubseteq Candy$ (*Icecream is candy.*)

(3) $Chocolate \sqsubseteq Candy$ (*Chocolate is candy.*)

(4) $Icecream \sqsubseteq \neg Chocolate$ (*Icecream is not chocolate.*)

(5) $Chocolate \sqsubseteq \neg Icecream$ (*Chocolate is not icecream.*)

(6) $Child \sqsubseteq \forall likes.Chocolate$ (*Children like only chocolate.*)

An appropriate solution for this example is to replace the overgeneralized definitions $\forall likes.Icecream$ and $\forall likes.Chocolate$ with their least common subsumer $\forall likes.Candy$. The resulting axiom $Child \sqsubseteq \forall likes.Candy$ claims that children like only candies without specifying the candy type:

**Example 6.4 (Rewritten Ex. 6.3).**

(1) $Child \sqsubseteq \forall likes.Candy$ (*Children like only candies.*)

(2) $Icecream \sqsubseteq Candy$ (*Icecream is candy.*)

(3) $Chocolate \sqsubseteq Candy$ (*Chocolate is candy.*)

(4) $Icecream \sqsubseteq \neg Chocolate$ (*Icecream is not chocolate.*)

(5) $Chocolate \sqsubseteq \neg Icecream$ (*Chocolate is not icecream.*)

The distinction between multiple and single overgeneralizations is non-trivial. A single overgeneralization occurs, if a definition of a concept $C$ captures typical rather than necessary features of $C$ and there are exceptions – subconcepts of $C$ contradicting the definition. In the case of multiple overgeneralizations, two or more definitions of the same concept are too specific and conflict with each other. Unfortunately, it seems to be impossible to define this distinction formally, since it is a matter of human expert interpretation.

From a practical perspective, it turns out that multiple overgeneralizations occur, if a concept has two or more incompatible definitions (cf. Ex. 6.3). This case has a certain structural similarity to the polysemy problem, where an unsatisfiable concept is subsumed by different disjoint concepts (cf. the tree example above). Practically, polysemy may be distinguished from multiple overgeneralizations by taking into account the level of abstraction of the disjoint concepts. In the case of polysemy, the disjoint superconcepts of the unsatisfiable concept usually occur in the upper structure of the taxonomy tree, whereas multiple overgeneralizations occur in lower levels of the taxonomy. However, this heuristic needs further investigation.

### 6.2.2 *Tracing Clashes*

The proposed approach to resolve logical inconsistencies is based on the tableau algorithm presented in Lam *et al.* (2006) for tracing clashes in unsatisfiable terminologies and rewriting problematic axioms. We adapt this tracing technique to the $\mathscr{ALCN}$ logic, which underlies the representation of ontologies considered in this book (see Sec. 4.4). The technique presented here is general enough to be used as the basis for developing similar algorithms for more expressive DLs. The proposed algorithm detects the relevant parts of the axioms that are responsible for logical contradictions.

Suppose that a TBox $\mathscr{T}$ contains axioms $\{\alpha_1,\ldots,\alpha_n\}$, where $\alpha_i$ refers to the axiom $A_i \sqsubseteq C_i$ or $A_i \equiv C_i$ $(i \in \{1,\ldots,n\})$. Checking the satisfiability of a concept description $A$, the tableau algorithm constructs a model of $A$ represented by a tree $\mathbf{T}$. Each node $x$ in this tree is labeled with a labeling set $\mathscr{L}(x)$ containing elements of the form $(a : C, I, a' : C')$, where $C$ and $C'$ are concept descriptions in negation normal form, $a$ and $a'$ are individual names, and $I$ is a set of axiom indices. An element $(a : C, I, a' : C')$ means that individual $a$ instantiates concept description $C$ due to the application of an expansion rule on $C'$, and $I$ contains the indices of the axioms, where $a : C$ originates from. $\mathbf{T}$ is initialized with root node $r$ and $\mathscr{L}(r)$ is initialized with $\{(a : A, \varnothing, nil)\}$.

The algorithm expands $\mathbf{T}$ according to the rules in Table 6.1.[9] It terminates if no expansion rules can be applied to tree nodes. The three rules $\equiv$+, $\equiv$– and $\sqsubseteq$ describe *unfolding*, i.e replacing defined names by their definitions. The rules $\equiv$+ and $\equiv$– reflect the symmetry of the definition $A \equiv C$, which is equivalent to $A \sqsubseteq C$ and $\neg A \sqsubseteq \neg C$. Rule $\sqsubseteq$ stands for the definitions of the form $A \sqsubseteq C$.

Unfolding a concept description containing disjunction results in non-deterministic expansion (Rule $\sqcup$): Two alternative labeling sets $\mathscr{L}(y)$ and $\mathscr{L}(z)$ are created. Only one of them has to be satisfiable for the terminology to be satisfiable.

For an existential role restriction (Rule $\exists$), the algorithm introduces a new individual $b$, which must satisfy a $C$ constraint. If an individual belongs to more than $n$ existential restrictions on $R$ and to a number restriction $\leq nR$, then it is necessary to merge the existing $R$-successors into $n$ individuals. If there are many merging combinations, then a contradiction arises only if every combination contains a clash. A solution is to choose one combination and to resolve clashes in it.

[9] Tags are provided to avoid circularity in the expansion of concept definitions. ”-” is a place holder. It stands for any value.

Table 6.1 Tableau expansion rules for tracing clashes in $\mathscr{ALCN}$ terminology.

| | |
|---|---|
| Rule $\equiv$+ | if $C_i \equiv D_i \in \mathscr{T}$, $(a : C_i, I, a' : A') \in \mathscr{L}(x)$, and $(a : C_i, I, a' : A')$ is not tagged, then $tag((a : C_i, I, a' : A'))$ and $\mathscr{L}(x) := \mathscr{L}(x) \cup \{(a : D_i, I \cup \{i\}, a : C_i)\}$ |
| Rule $\equiv$– | if $C_i \equiv D_i \in \mathscr{T}$, $(a : \neg C_i, I, a' : A') \in \mathscr{L}(x)$, and $(a : \neg C_i, I, a' : A')$ is not tagged, then $tag((a : \neg C_i, I, a' : A'))$ and $\mathscr{L}(x) := \mathscr{L}(x) \cup \{(a : \neg D_i, I \cup \{i\}, a : \neg C_i)\}$ |
| Rule $\sqsubseteq$ | if $C_i \sqsubseteq D_i \in \mathscr{T}$, $(a : C_i, I, a' : A') \in \mathscr{L}(x)$ and $(a : C_i, I, a' : A')$ is not tagged, then $tag((a : C_i, I, a' : A'))$ and $\mathscr{L}(x) := \mathscr{L}(x) \cup \{(a : D_i, I \cup \{i\}, a : C_i)\}$ |
| Rule $\sqcap$ | if $(a : C_1 \sqcap C_2, I, a' : A') \in \mathscr{L}(x)$, then $\mathscr{L}(x) := \mathscr{L}(x) \cup \{(a : C_1, I, a : C_1 \sqcap C_2)\} \cup \{(a : C_2, I, a : C_1 \sqcap C_2)\}$ |
| Rule $\sqcup$ | if $(a : C_1 \sqcup C_2, I, a' : A') \in \mathscr{L}(x)$, then $\mathscr{L}(y) := \mathscr{L}(x) \cup \{(a : C_1, I, a : C_1 \sqcup C_2)\}$ and $\mathscr{L}(z) := \mathscr{L}(x) \cup \{(a : C_2, I, a : C_1 \sqcup C_2)\}$ |
| Rule $\exists$ | if $(a : \exists R.C, I, a' : A') \in \mathscr{L}(x)$, then $\mathscr{L}(x) := \mathscr{L}(x) \cup \{(b : C, I, a : \exists R.C)\}$, where $b$ is a new individual name not occurring in $\mathscr{L}(x)$ |
| Rule $\leq$ | if $(a :\leq nR, I, a' : A') \in \mathscr{L}(x)$ and $(b_i : C_i, -, a : -) \in \mathscr{L}(x)$ $(i \in \{1, \ldots, n+1\})$, where $b_1, \ldots, b_{n+1}$ are distinct individuals, then for each pair $b_i, b_j$ $(i, j \in \{1, \ldots, n+1\}, i \neq j)$ $\mathscr{L}(x_{i,j})$ is obtained from $\mathscr{L}(x)$ by replacing each occurrence of $b_i$ by $b_j$ |
| Rule $\geq$ | if $(a :\geq nR, I, a' : A') \in \mathscr{L}(x)$, and the above rules cannot be applied, then if there is no $(b : C, J, a : \exists R.C) \in \mathscr{L}(x)$, then $\mathscr{L}(x) := \mathscr{L}(x) \cup \{(b : \top, I, a :\geq nR)\}$, where $b$ is a new individual name |
| Rule $\forall$ | if $(a : \forall R.C, I, a' : A') \in \mathscr{L}(x)$, and the above rules cannot be applied, for each $(b : D, -, a : -) : \mathscr{L}(x) := \mathscr{L}(x) \cup \{(b : C, I, a : \forall R.C)\}$ |

Rule $\geq$ introduces a new $R$-successor as representative, if none was introduced yet. Rule $\forall$ imposes a new $C$ constraint on existing $R$-successors.

The following elements/pairs of elements contained in a labeling set are called *clashes*:

(a) $(a : C, -, -)$ and $(a : \neg C, -, -)$,

(b) $(a :\leq nR, -, -)$ and $(a :\geq mR, -, -)$, for $n < m$,

(c) $(a : \bot, -, -)$.

If a new element $e = (a : C, -, -)$ is added to $\mathscr{L}(x)$, then the algorithm checks, whether $e$ introduces a new clash, in which $a$ is involved. If this is the case, then the pair $(e, e')$ ($e'$ is another element from $\mathscr{L}(x)$ constituting the clash) is added to the set $Clashes(\mathscr{L}(x))$.

The algorithm returns a model tree, i.e. a number of labeling sets $\mathscr{L}(x)$, and a set of clashes. Only one labeling set has to be repaired for the terminology to be satisfiable.

The given algorithm extends the canonical tableau algorithm for $\mathscr{ALCN}$ (see Sec. 4.4) by adding unfolding rules and labeling sets. Thus, properties of the original algorithm (decidability, complexity) are relevant for the presented algorithm (see Sec. 4.4).

In order to account for problematic expansions, we introduce the notion of *trace* as defined below.

**Definition 6.1 (Trace).** Given an element $e = (a_0 : C_0, I_0, a_1 : C_1)$ in a labeling set $\mathscr{L}(x)$, the trace of $e$, $Trace(e)$, is a sequence of the form $\langle (a_0 : C_0, I_0, a_1 : C_1), (a_1 : C_1, I_1, a_2 : C_2), \ldots, (a_{n-1} : C_{n-1}, I_{n-1}, a_n : C_n), (a_n : C_n, \varnothing, nil) \rangle$, where $I_{i-1} \subseteq I_i$ for each $i \in \{1, \ldots, n\}$ and every element in the sequence belongs to $\mathscr{L}(x)$.

### 6.2.3 *Resolving Overgeneralization*

As already mentioned above, multiple overgeneralizations can be repaired by replacing conflicting definitions with their least common subsumer, which is defined as follows.

**Definition 6.2 (Least common subsumer).** Given a collection of $C_1, \ldots, C_n$ concept descriptions, the least common subsumer (lcs) of $C_1, \ldots, C_n$ is the most specific concept description that subsumes $C_1, \ldots, C_n$, i.e. it is a concept description $D$ such that

(1) $\forall i \in \{1, \ldots, n\} : C_i \sqsubseteq D$ ($D$ is a common subsumer);

(2) if there is a concept description $E$ such that
$\forall i \in \{1, \ldots, n\} : C_i \sqsubseteq E$ then $D \sqsubseteq E$ ($D$ is least).

In the DL community, the procedure for computing *lcs* based on the tableau algorithm is well-developed. We do not replicate this work here; the interested reader is referred to Donini *et al.* (2009).

Let us now examine the problem of resolving the case of single overgeneralization. Given a clash $(a_1 : C_1, I_1, a'_1 : C'_1), (a_2 : C_2, I_2, a'_2 : C'_2)$, the union of the index sets $I_1$ and $I_2$ gives the set of axioms, which cause the clash. This set of axioms corresponds to the *minimal unsatisfiability-preserving sub-TBox* (MUPS) as defined in (Lam *et al.*, 2006). Lam *et al.* (2006) show that removing[10] one of the concepts appearing in the clash traces from the appropriate parts of axioms in $I$ is sufficient to resolve the clash. This result is replicated in the following lemma.

[10]Removal of $C$ means replacement of $C$ with $\top$.

**Lemma 6.1 (Lam, 2006).** *Let $D$ be the set of all concepts appearing in the traces of the clash elements $(a_1 : C_1, I_1, a'_1 : C'_1)$ and $(a_2 : C_2, I_2, a'_2 : C'_2)$. Removing one of the concepts in $D$ from one of the axioms in $I_1 \cup I_2$ is sufficient to resolve the clash.*

***Proof.*** For any concept in $D$, it occurs in the traces and has adjacent elements, which are before or after. For any two adjacent elements $e_1$ and $e_2$ in a trace, there are only two options:

(1) $e_1$ and $e_2$ have the form of $(a : E_1, -, a : E_2)$ and $(a : E_2, -, -)$ containing the same individual $a$. This means the concept $E_1$ subsumes $E_2$. If $E_1$ or $E_2$ is removed, the subsumption relationship between $E_1$ and $E_2$ is removed. Therefore, the individual $a$ does not belong to $E_1$, nor does it belong to any of the concepts in the elements preceding the occurrence of $e_1$ in the traces. Thus, the clash is resolved.

(2) $e_1$ and $e_2$ have the form of $(a : E_1, -, b : E_2)$ and $(b : E_2, -, -)$ containing different individuals $a$ and $b$. This means that there is a role restriction $E_2$ imposing the $E_1$ constraint. If $E_1$ or $E_2$ is removed, then there is no such individual $a$ participating in the role, and all the concepts in the elements preceding the occurrence of $e_1$ are not related to $a$. Thus, the clash is resolved.

□

One of the most important practical problems is selecting, which of the problematic axioms needs to be modified. Many ranking criteria were suggested in the literature on ontology debugging (Schlobach and Cornet, 2003; Haase *et al.*, 2005; Kalyanpur, 2006; Lam *et al.*, 2006):

- *Arity* of an axiom $\alpha$ denotes in how many clashes $\alpha$ is involved. The higher the arity is, the lower is the rank of $\alpha$.
- *Semantic impact* of $\alpha$ denotes how many entailments are lost if $\alpha$ is removed. Axioms with a high semantic impact are ranked higher.
- *Syntactic relevance* denotes how often concept and role names occurring in an axiom $\alpha$ are used in other axioms in the ontology. Axioms containing elements that are frequently occurring in the ontology are ranked higher.
- *Manual ranking* of $\alpha$ can be provided by the ontology engineer.
- *Frequency ranking* of $\alpha$ is used in approaches to semi-automatic ontology extraction and denotes how often concepts and roles in $\alpha$ occur in external data sources.

Here, we do not discuss ranking strategies (because they are application-specific) and assume that any of the strategies mentioned above can be applied. Suppose a concept description $C$ is chosen to be removed from an axiom $\alpha$ in $\mathscr{T}$ to resolve a clash (or clashes). The next questions is: How can we find an appropriate concept description $C'$ that can resolve the clash by replacing $C$? We are looking for a replacement that resolves the clash, does not cause new clashes or entailments, and preserves as many entailments implied by $\mathscr{T}$ as possible. Definition 6.3 defines such a replacement.

**Definition 6.3 (Minimal nonconflicting subsumer).** The following is given: a terminology $\mathscr{T}$, a concept description $C$ that is chosen to be removed from an axiom $\alpha \in \mathscr{T}$. Let an axiom $\alpha'$ be obtained from $\alpha$ by replacing $C$ with a concept description $C'$, $\mathscr{T}' := \mathscr{T} \setminus \{\alpha\} \cup \{\alpha'\}$, an axiom $\alpha''$ be obtained from $\alpha$ by replacing $C$ with $\top$, and $\mathscr{T}'' := \mathscr{T} \setminus \{\alpha\} \cup \{\alpha''\}$. $C'$ is a minimal nonconflicting subsumer (MNS) of $C$ if the following conditions hold:

(1) The set of clashes in all possible labeling sets constructed for $C$ towards $\mathscr{T}'$ contains the same number of clashes as contained in all possible labeling sets constructed for $C$ towards $\mathscr{T}''$.
(2) If $C' \neq \top$, then there exists an entailment $\beta$, such that $\mathscr{T} \models \beta$, $\mathscr{T}'' \not\models \beta$, and $\mathscr{T}' \models \beta$.
(3) There exists no entailment $\beta$, such that $\mathscr{T} \not\models \beta$ and $\mathscr{T}' \models \beta$.
(4) There exists no concept description $C''$ with the same properties such that $C''$ preserves more entailments from $\mathscr{T}$.

Condition (1) guarantees that MNS resolves the clashes where $\alpha$ and $C$ are involved and does not introduce new clashes. Due to (2) MNS preserves at least one entailment from $\mathscr{T}$ that will be lost with the removal of $C$. Condition (3) excludes new entailments that are not implied by $\mathscr{T}$, (4) guarantees that MNS preserves as much information as possible. Obviously, $MNS(C)$ should subsume $C$; otherwise, the replacement will introduce new entailments. Lemma 6.2 below shows how to calculate MNS.

Before passing on to Lemma 6.2, we need to introduce one more abbreviation. Suppose a number restriction $\leq nR$ causes impossible mergings of R-successors. For example, $\mathscr{T} = \{A \sqsubseteq \leq 1R \sqcap \exists R.B_1 \sqcap \exists R.B_2, B_1 \equiv \neg B_2\}$. Contradicting R-successors imply $\geq mR$ number restriction with $n < m$. In the example above, $A \sqsubseteq \geq 2R$ is implied. In order to induce $m$ from $\exists R.C_1 \sqcap \ldots \sqcap \exists R.C_k$, one needs to detect, which $C_{i \in \{1,\ldots,k\}}$ are disjoint (see Brandt *et al.*, 2002). An at-least number restriction on $R$ induced from a concept description $C$ is denoted by $min_R(C)$. Brandt *et al.* (2002) show how $min_R(C)$ can be computed in polynomial time. We do not replicate this work here.

**Lemma 6.2.** *The following is given: a terminology $\mathscr{T}$, an unsatisfiable concept UC, a labeling set $\mathscr{L}(x)$ constructed for UC towards $\mathscr{T}$, a concept description C that is chosen to be removed from an axiom $\alpha \in \mathscr{T}$. A concept description $C'$ is MNS of C iff the following conditions hold*[11]*:*

*(1) If $C \doteq\ \geq mR$ and $(a : \geq mR, -, -), (a : \leq nR, -, -) \in \mathscr{L}(x)$ for $n < m$, then $C' \doteq\ \geq nR$.*

*(2) Else if $C \doteq\ \leq nR$ and $(a : \geq mR, -, -), (a : \leq nR, -, -) \in \mathscr{L}(x)$ for $n < m$, then $C' \doteq\ \leq mR$.*

*(3) Else if $C \doteq\ \leq nR$ and $(a : \leq nR, -, -) \in \mathscr{L}(x)$, then $C' \doteq\ \leq kR$, where $k = min_R(\bigsqcap_{(b:D,-,a:\exists R.D) \in \mathscr{L}(x)} \exists R.D)$.*

*(4) Else if $C \doteq \bot$ or $C \doteq \forall R.\bot$, then $C' \doteq \top$.*

*(5) Else if $C \doteq \forall R.D$ or $C \doteq \exists R.D$, then $C' \doteq \forall R.MNS(D)$ or $C' \doteq \exists R.MNS(D)$ correspondingly.*

*(6) Else $C' \doteq C_1 \sqcap \ldots \sqcap C_n$, such that for all $i \in \{1, \ldots, n\}$*

*(a) $(a : C, -, -)$ is the first element in the trace of $(a : C_i, -, -)$ and for every clash element e such that C occurs in $Trace(e)$: $C_i$ is not in $Trace(e)$*

*or*

*(b) there exists D such that $(a : C, -, -)$ is the first element in the trace of $(a : D, -, -)$ and there is a clash element e such that C and D both occur in $Trace(e)$ and $C_i \doteq MNS(D)$;*

*If there is no such $C'$ then $C' := \top$.*

***Proof.***

(1) The concept description $\geq mR$ has no other subsumers except $\geq kR$ for $k < m$. Therefore $\geq nR$ is the only possible MNS for $C$.

(2) $C \doteq\ \leq nR$ can be subsumed not only by $\leq kR$ for $k > n$, but also by some concept description $D \doteq \exists R.D_1 \sqcap \ldots \sqcap \exists R.D_l$ with $min_R(D) = k$ for $k > n$. The replacement of $C$ with $D$ will introduce new entailments not implied by $\mathscr{T}$. Hence, $\leq mR$ is the only possible MNS for $C$.

(3) $C \doteq\ \leq nR$ is involved in a clash if it causes an impossible merging of $R$-successors of $a$. Hence, the relational restrictions defined on $a$ imply a number restriction $\geq kR$ conflicting with $\leq nR$, due to $n < k$. Thus, $\leq kR$ is the MNS of $C$.

(4) Trivial.

[11]The symbol $\doteq$ denotes the syntactical equality of concept descriptions.

(5) Trivial.

(6) Every $C_i$ is either an immediate subsumer of $C$ not occurring in its clash trace (a) or a MNS of an immediate subsumer $D$ of $C$ occurring in its clash trace (b). $\square$

For every type of concept description $C$, MNS($C$) is defined in a deterministic way. Therefore, for any concept description there is only one MNS.

Let us reconsider Ex. 6.1. The model tree for the concept *Penguin* in this example consists of the following elements:

$$\begin{aligned}\mathscr{L}(x) \quad = \quad & \{(a : Penguin, \varnothing, nil), (a : Bird \sqcap \neg CanFly, \{4\}, a : Penguin), \\ & (a : Bird, \{4\}, a : Bird \sqcap \neg CanFly), \\ & (a : \neg CanFly, \{4\}, a : Bird \sqcap \neg CanFly), \\ & (a : CanFly, \{4, 1\}, a : Bird), (a : CanMove, \{4, 1, 2\}, a : CanFly)\}\end{aligned}$$

Thus, there is a clash $(a : \neg CanFly, \{4\}, a : Penguin)$, $(a : CanFly, \{4, 1\}, a : Bird)$ and the set of problematic axioms is $\{1, 4, 5\}$. Suppose the concept *CanFly* is chosen to be removed from axiom 1. According to Lemma 6.2, $CanMove$ is MNS of $CanFly$. If $CanFly$ is replaced by $CanMove$ in axiom 1, then the entailments of the form $X \sqsubseteq CanFly$, where $X$ is a subconcept of *Bird* (for example, *Canary*), would be lost. Such situations are undesirable, because the clash ($CanFly, \neg CanFly$) concerns only the conflict between the overgeneralized concept *Bird* and the exception *Penguin*. In order to keep the entailments, we suggest to introduce a new concept $FlyingBird$ to the terminology, which will capture the original meaning of $Bird$.

### 6.2.4 *Root and Derived Concepts*

The result of the application of the proposed debugging method is dependent on the order the input concepts. Suppose a terminology contains the following axioms:

(1) $A \sqsubseteq B$

(2) $B \sqsubseteq C$

(3) $C \sqsubseteq D$

(4) $C \sqsubseteq \neg D$

The problem is obviously caused by contradicting axioms (3) and (4). However, if the inconsistent concept $A$ will be debugged before $C$, then axiom (1) or (2) can be rewritten instead, which depends on the selected ranking strategy. Unsatisfiable concepts can be divided into two classes:

1. *Root concepts* are atomic concepts, for which a clash found in their definitions does not depend on a clash of another atomic concept in the ontology;
2. *Derived concepts* are atomic concepts, for which a clash found in their definitions either directly (via explicit assertions) or indirectly (via inferences) depends on a clash of another atomic concept (Kalyanpur, 2006).

In order to debug a derived concept (concept $A$ in the example above), it is enough to debug corresponding root concepts (concept $C$ in the example above). Thus, it is reasonable to debug the root concepts only. The technique of distinguishing between root and derived concepts was proposed by Kalyanpur (2006). We integrate this technique into the presented approach.

**Definition 6.4 (Minimal clash-preserving sub-TBox MCPS).** Let $C$ be a concept, for which its model tree obtained relative to a terminology $\mathscr{T}$ contains clashes. A sub-TBox $\mathscr{T}' \subseteq \mathscr{T}$ is a MCPS of $C$, if a model tree for $C$ towards $\mathscr{T}'$ contains clashes and a model tree of $C$ towards every $\mathscr{T}'' \subset \mathscr{T}'$ contains no clashes.

The union of the axiom indices in $I$ of all clash elements in the model tree of $C$ corresponds to MUPS of $C$ in (Schlobach and Cornet, 2003) and constitutes MCPS of $C$ in (Lam *et al.*, 2006). Given a specific clash $(e_1, e_2)$, $\mathrm{MCPS}_{(e_1,e_2)}(C)$ is defined similarly as in Definition 6.4 except for $\mathrm{MCPS}_{(e_1,e_2)}(C)$ preserves only the clash $(e_1, e_2)$.

**Definition 6.5 (Root and derived concepts).** An atomic concept $C$ is **derived** from an atomic concept $C'$ if there exists a clash $(e_1, e_2)$ in the model tree of $C$ such that $\mathrm{MCPS}_{(e_1,e_2)}(C')$ is a subset of $\mathrm{MCPS}_{(e_1,e_2)}(C)$. If there is no concept $C'$, from which $C$ is derived, then $C$ is a **root** concept.

Proposition 6.1 shows how to find dependencies between problematic concepts.

**Proposition 6.1.** *Given a clash $(e_1, e_2)$ in a labeling set $\mathscr{L}(x)$ constructed for a concept $C$, $C$ is derived from a concept $C'$ (towards this clash) if and only if* $\exists a : (a : C', -, -) \in Trace_{\mathscr{L}(x)}(e_1)$ *and* $(a : C', -, -) \in Trace_{\mathscr{L}(x)}(e_2)$.

$\mathrm{MCPS}_{(e_1,e_2)}(C')$ is a subset of $\mathrm{MCPS}_{(e_1,e_2)}(C)$ if and only if the model tree for $C'$ is a subset of the model tree for $C$ and both of these trees contain the clash $(e_1, e_2)$ (modulo differences in the axiom indices set). Obviously, $C'$ is in the traces of the clash elements in its own model tree. Therefore, if $C$ is derived from $C'$ towards $(e_1, e_2)$, then $C'$ is in the traces of $e_1$ and $e_2$ in $\mathscr{L}(x)$. On the other hand, if $C'$ occurs in the traces of $e_1$ and $e_2$, then the model tree of $C'$ contains this clash and is a subset of $\mathscr{L}(x)$.

The distinction between root and derived concepts can be used in order to ensure that only those concepts, which are defined inconsistently, are considered.

### 6.2.5 *Rewriting Algorithm*

The algorithm for repairing a terminology $\mathscr{T}$ containing an unsatisfiable root concept $C$ is shown in Algorithm 6.1. First, the algorithm checks whether $C$ has two (or more) conflicting definitions ($D$ and $D'$). In this case, polysemy or multiple overgeneralization is suspected. If the user confirms multiple overgeneralization then the algorithm replaces the problematic definitions with their least common subsumer. Polysemy should be resolved by an external procedure.

If there are no multiple conflicting definitions for $C$ then the algorithm resolves single overgeneralization using the model tree for $C$. First, a labeling set $\mathscr{L}(x)$ in the model tree is chosen to be repaired according to a given ranking (see description of possible rankings in Sec. 6.2.3). All clashes in $\mathscr{L}(x)$ are repaired as follows. For every clash $(e_1, e_2)$ a concept description $X$ from the trace of a clash element $e_{k\in\{1,2\}}$ is chosen to be rewritten in an axiom $\alpha$ ($\alpha \doteq A_i \sqsubseteq B_i$ or $\alpha \doteq A_i \equiv B_i$) according to the ranking. $X$ is replaced with its MNS in $\alpha$. If definition $B_i$ of $A_i$ is inconsistent towards $\mathscr{T}$, then part $X$ of $B_i$ is replaced with its MNS. Otherwise, a new concept $A^{new}$ is introduced to capture the original semantics of $A_i$. The name $A^{new}$ is constructed automatically from the original name $A$ and the problematic concept description $X$.

This change of semantics of $A_i$ influences all its subconcepts occurring in the trace of the clash element $e_k$. The set $T$ consists of elements from $\mathscr{L}(x)$ that are contained in the trace of the clash element $e_k$ between the unsatisfiable concept $C$ and the rewritten concept description $X$. The recursive *split&replace* subroutine a) for all axioms not occurring in the trace of the clash element $e_k$ replaces the original concept $A$ with the newly introduced concept $A^{new}$, b) splits atomic concepts that are involved in the clash. Concepts appearing in the trace of the clash element $e$ earlier are split first.

**Algorithm 6.1** Repair terminology $\mathscr{T}$ containing unsatisfiable concept $C$.

**Require:** terminology $\mathscr{T}$, concept $C$ unsatisfiable towards $\mathscr{T}$

**if** $\exists \alpha, \alpha' \in \mathscr{T} : (\alpha \doteq C \sqsubseteq D$ **or** $\alpha \doteq C \equiv D)$ **and** $(\alpha' \doteq C \sqsubseteq D'$ **or** $\alpha' \doteq C \equiv D')$ **and** $D, D'$ are consistent towards $\mathscr{T}$ **and** $D \sqcap D' \sqsubseteq \bot$ towards $\mathscr{T}$ **then**

  report polysemy **or** multiple overgeneralization

  **if** multiple overgeneralization **then**

    remove $\alpha, \alpha'$ from $\mathscr{T}$ **and** add $C \sqsubseteq lcs(D, D')$ to $\mathscr{T}$

  **else if** polysemy **then**

    external repair $\alpha, \alpha'$

  **end if**

**else**

  choose a set $\mathscr{L}(x)$ of the model tree for $C$ according to *ranking*

  **for** clashes $(e_1, e_2)$ in the set $\mathscr{L}(x)$ **do**

    choose concept description $X$ occurring in axiom $\alpha$ ($\alpha \doteq A_i \sqsubseteq B_i$ **or** $\alpha \doteq A_i \equiv B_i$), such that for some $e_{k \in \{1,2\}} : (x : X, \{i, \ldots\}, -) \in Trace(e_k)$, to be rewritten according to *ranking*

    **if** $B_i$ is inconsistent towards $\mathscr{T}$ **then**

      replace $X$ with $MNS(X)$ in $\alpha$

    **else**

      remove $\alpha$ from $\mathscr{T}$

      $\alpha^{new}$ is obtained from $\alpha$ by replacement of $X$ with MNS($X$)

      add $\alpha^{new}$ to $\mathscr{T}$

      add $A^{new} \sqsubseteq A_i$ and $A^{new} \sqsubseteq B_i$ to $\mathscr{T}$

      **let** $T$ be a subsequence of $Trace(e_k)$ between the elements $(b : X, -, -)$ and $(-, -, a : C)$ (not inclusive)

      *split&replace*($A$, $A^{new}$, $T$)

    **end if**

  **end for**

**end if**

**Subroutine** *split&replace*

**Require:** $A, A^{new}, T$

**for** $\alpha \in \mathcal{T}$ $(\alpha \doteq E_i \sqsubseteq F_i$ **or** $\alpha \doteq E_i \equiv F_i)$ **do**
    **if** $A$ occurs in $F_i$ `and` $i$ does not occur in $T$ **then**
        replace $A$ by $A^{new}$ in $\beta$
    **end if**
**end for**
$(b : B', -, a : B)$ is the **next** element of $T$
**if** $B$ is atomic **then**
    $B''$ is obtained by replacing $A$ with with $A^{new}$ in $B'$
    **if** $B \sqsubseteq B' \in \mathcal{T}$ **then**
        add $B^{new} \sqsubseteq B''$ to $\mathcal{T}$
    **else if** $B \equiv B' \in \mathcal{T}$ **then**
        add $B^{new} \equiv B''$ to $\mathcal{T}$
    **end if**
**end if**
*split&replace*$(B, B^{new}, T)$

**Example 6.5.**

| | |
|---|---|
| (1) $Pilot \sqsubseteq \forall aviates.Airplane$ | (*Pilots aviate only airplanes.*) |
| (2) $Airplane \sqsubseteq Transport$ | (*An airplane is a transport.*) |
| (3) $PassengerPlane \sqsubseteq Airplane$ | (*A passenger plane is an airplane.*) |
| (4) $FighterPilot \sqsubseteq Pilot$ | (*A fighter pilot is a pilot.*) |
| (5) $FighterPilot \sqsubseteq \exists aviates.FightingMachine$ | (*Fighter pilots sometimes aviate fighting machines.*) |
| (6) $FightingMachine \sqsubseteq \neg Transport$ | (*A fighting machine is not a transport.*) |

Example 6.5 shows the application of the algorithm. The concept *FighterPilot* is unsatisfiable because it is subsumed by $\forall aviates.Transport$ and $\exists aviates.\neg Transport$. The label tree for $FighterPilot$ contains the following elements:

$$
\begin{aligned}
\mathscr{L}(x) = \quad & \{(a : FighterPilot, \{\}, nil) \\
& (a : \exists aviates.FightingMachine, \{5\}, a : FighterPilot) \\
& (b : FightingMachine, \{5\}, a : \exists aviates.FightingMachine) \\
& (b : \neg Transport, \{5,6\}, b : FightingMachine) \\
& (a : Pilot, \{4,5\}, a : FighterPilot) \\
& (a : \forall aviates.Airplane, \{1,4,5\}, a : Pilot) \\
& (b : Airplane, \{1,4,5\}, \forall aviates.Airplane) \\
& (b : Transport, \{1,2,4,5\}, b : Airplane)\ \}
\end{aligned}
$$

The clash elements are $e_1 = (b : \neg Transport, \{5,6\}, b : FightingMachine)$ and $e_2 = (b : Transport, \{1,2,4,5\}, b : Airplane)$. The trace of $e_1$ is

$$
\begin{aligned}
& \{(b : \neg Transport, \{5,6\}, b : FightingMachine), \\
& (b : FightingMachine, \{5\}, a : \exists aviates.FightingMachine), \\
& (a : \exists aviates.FightingMachine, \{5\}, a : FighterPilot), \\
& (a : FighterPilot, \{\}, nil)\}.
\end{aligned}
$$

The trace of $e_2$ is

$$
\begin{aligned}
& \{(b : Transport, \{1,2,4,5\}, b : Airplane), \\
& (b : Airplane, \{1,4,5\}, \forall aviates.Airplane), \\
& (a : \forall aviates.Airplane, \{1,4,5\}, a : Pilot), \\
& (a : Pilot, \{4,5\}, a : FighterPilot), \\
& (a : FighterPilot, \{\}, nil)\}.
\end{aligned}
$$

The set of problematic axioms is $\{1,2,4,5,6\}$.The problem is caused by axiom (2), which is too general stating that every airplane is a transport. Axiom (6) stating that a fighting machine is not a transport represents an exception contradicting axiom (2).

Suppose the concept $Transport$ in axiom (2) is chosen to be rewritten. Since the concept $Transport$ has no other subsumer except for $\top$, the new axiom obtained from axiom (2) through the replacement of $Transport$ with $\top$ is $Airplane \sqsubseteq \top$, which is equivalent to the removal of axiom (2). The original meaning of $Airplane$ is captured by the two new axioms: $TransportAirplane \sqsubseteq Airplane$ and $TransportAirplane \sqsubseteq Transport$. The new concept name ($A^{new}$ in the algorithm, $TransportAirplane$ in this example) is constructed automatically.

Based on the trace of $e_2$, the set $T$ contains the following elements:

$(b : Airplane, \{1,4,5\}, \forall aviates.Airplane)$,

$(a : \forall aviates.Airplane, \{1,4,5\}, a : Pilot)$

$Transport$, $TransportAirplane$, and $T$ are further passed to the **split&replace** subroutine. The subroutine first concerns all axioms from $\mathscr{T}$, which do not occur in $T$ and replaces all $Transport$ occurrences with $TransportAirplane$ in these axioms. In our example, it concerns axiom (3).

Then, the subroutine goes through the elements of $T$ and splits atomic concepts involved in the clash. $(a : \forall aviates.Airplane, \{1,4,5\}, a : Pilot)$ is the next element of $T$ such that $Pilot$ is atomic. Since the axiom $Pilot \sqsubseteq \forall aviates.Airplane$ belongs to $\mathscr{T}$, the new axioms $AviatesTransportAirplanePilot \sqsubseteq \forall aviates.TransportAirplane$ and $AviatesTransportAirplanePilot \sqsubseteq Pilot$ are added to $\mathscr{T}$. The concept name $AviatesTransportAirplanePilot$ is constructed automatically. The rewritten terminology is shown below.

**Example 6.6 (Rewritten Ex. 6.5).**
*(1)* $Pilot \sqsubseteq \forall aviates.Airplane$
*(2)* $PassengerPlane \sqsubseteq TransportAirplane$
*(3)* $FighterPilot \sqsubseteq Pilot$
*(4)* $FighterPilot \sqsubseteq \exists aviates.FightingMachine$
*(5)* $FightingMachine \sqsubseteq \neg Transport$
*(6)* $TransportAirplane \sqsubseteq Airplane$
*(7)* $TransportAirplane \sqsubseteq Transport$
*(8)* $AviatesTransportAirplanePilot \sqsubseteq Pilot$
*(9)* $AviatesTrasportAirplanePilot \sqsubseteq \forall aviates.TrasportAirplane$

It is easy to see that the proposed algorithm extends the semantics of the “split” concepts, whereas the semantics of other concepts remains unchanged.

### 6.2.6 *Prototypical Implementation*

In (Ovchinnikova *et al.*, 2007), a prototypical implementation of the idea of splitting overgeneralized concepts in OWL Lite ontologies is discussed. In order to make Description Logic inferences available to the algorithm, we integrated into our system the *KAON2* DL reasoner[12] designed for managing of and reasoning on OWL ontologies.

As a base ontology we took the famous wine ontology[13] describing different sorts of wine, grapes, and wine regions. This ontology was created manually and is known not to contain any inconsistencies.

[12] *http://kaon2.semanticweb.org*
[13] *http://www.w3.org/TR/owl-guide/wine.owl*

In order to introduce logical contradictions, we automatically extracted additional axioms from a collection of domain related texts. First, we generated a document set, which was automatically crawled from the web with the *BootCat Tools* (Baroni and Bernardini, 2004), using the vocabulary of the wine ontology as seed terms. We thus obtained a domain corpus of 288 documents comprising 182 754 tokens. This corpus served as input to the ontology extraction step. For this purpose we have used the freely available *Text2Onto* tool[14], developed at the AIFB (Karlsruhe, Germany), because this tool is capable of extracting not only basic relations such as taxonomy, but also disjointness and equivalence (Cimiano and Völker, 2005).

In the hereby automatically generated ontology, we however found only concepts (2155), instances (986), and subclass (385) and instance (211) relations. We then manually filtered the extracted relations to exclude errors. We also manually reformatted some relations to avoid a syntactic mismatch with the original ontology. This finally resulted in an ontology of 137 valid subclass relations and 83 instance relations.

The generated ontology proved to contain several logical contradictions with respect to the original wine ontology. For each inconsistent concept, Algorithm 6.1 described above found a set of axioms causing the problem and suggested an error class (multiple overgeneralization/polysemy or single overgeneralization). We manually confirmed several cases of polysemy as, for example, the *champagne* class being subclass of both *region* and *wine* or the *pinot noir* class, which was defined to be a subclass of *wine* and *grape*.

Several cases of single overgeneralization were detected automatically. For example, the class *LateHarvest* originally defined to be a sweet wine was claimed to be overgeneralized after an exception *RieslingSpaetlese* was added, which was defined to be a late harvest wine and a dry wine. Axioms to be rewritten were selected manually. The developed debugging program correctly rewrote all selected axioms. For example, axiom (1) in the ontology fragment 6.7 was rewritten as shown below.

**Example 6.7.**

Original ontology fragment:

(1) $LateHarvest \sqsubseteq Wine \sqcap \forall hasSugar.Sweet$

(2) $Dry \sqsubseteq WineSugar \sqcap \neg Sweet$

(3) $Sweet \sqsubseteq WineSugar \sqcap \neg Dry$

(4) $Dry \sqsubseteq WineSugar \sqcap \neg Sweet$

(5) $RieslingSpaetlese \sqsubseteq LateHaevest \sqcap Riesling \sqcap \forall hasSugar.Dry$

[14] *http://ontoware.org/projects/text2onto/*. Special thanks to Johanna Völker, who gave us helpful support.

Rewritten ontology fragment:

(1) $LateHarvest \sqsubseteq Wine \sqcap \forall hasSugar.WineSugar$

(2) $Dry \sqsubseteq WineSugar \sqcap \neg Sweet$

(3) $Sweet \sqsubseteq WineSugar \sqcap \neg Dry$

(4) $Dry \sqsubseteq WineSugar \sqcap \neg Sweet$

(5) $RieslingSpaetlese \sqsubseteq LateHaevest \sqcap Riesling \sqcap \forall hasSugar.Dry$

(6) $SweetLateHarvest \sqsubseteq Wine \sqcap \forall hasSugar.Sweet$

## 6.3 Concluding Remarks

This chapter concerns ensuring consistency of the developed KB, which is crucial for drawing correct inferences. In Sec. 6.1, we investigated the problem of conceptual consistency of frame networks with respect to natural language understanding and proposed a methodology for cleaning up frame relations. The methodology is developed using examples of frame relations contained in FrameNet. However, it is applicable to any frame-like relational network. The methodology consists in assigning ontological types to situations and entities, to which frames and frame roles refer, and ensuring consistency of the relations defined on frames and frame roles using constraints formulated in Sec. 6.1.

The inconsistency repair procedure is manual, because it presupposes a deep understanding of the ontological nature of the frames under consideration. A future work direction concerns development of methods for automatic detection of conceptual inconsistencies. For example, contrasting different resources of knowledge such as WordNet and FrameNet, similar to what is proposed by Verdezoto and Vieu (2011), can help to spot conceptual errors with respect to such relations as parthood or causation, which are present in both resources.

Section 6.2 concerns the issue of logical inconsistency of the axioms formalized using a syntactically rich representation language. In the case of the proposed integrative knowledge base, this issue concerns the ontological module only. The presented approach to repair of inconsistent ontologies is an integration of ideas proposed in (Lam *et al.*, 2006) and (Ovchinnikova and Kühnberger, 2006). The algorithm detects problematic axioms that cause a logical contradiction, distinguishes between different types of logical contradictions, and repairs the ontology. This approach is knowledge preserving in the sense that it keeps as many non-conflicting knowledge contained in the original ontology as possible.

The presented algorithm is defined for the $\mathcal{ALCN}$-DL. Since it is based on a tableau algorithm, it can be easily extended for more expressive versions of DL, for which a tableau algorithm has been defined, see Baader and Sattler (2001) for an overview of such DLs. The developed procedure is not fully automatic, because it presupposes a manual distinction between polysemy and multiple overgeneralization. Heuristics can be developed in order to automatize this distinction (e.g., the level of abstraction of the problematic definitions). This issue needs further investigation.

## Chapter 7

# Abductive Reasoning with the Integrative Knowledge Base

This chapter is concerned with extensions of the abductive inference procedure implemented in the reasoning system *Mini-TACITUS* (Mulkar *et al.*, 2007). The extensions are intended to make the system able to reason with the developed integrative knowledge base.

Since *Mini-TACITUS* was not originally designed for large-scale processing, it was necessary to perform several optimization steps, so that the system could treat a large knowledge base. The performed optimization steps are described in Sec. 7.1 of this chapter. Section 7.2 focuses on the solutions to two pragmatic problems encountered in the application of weighted abduction to NLU. Sections 7.3 and 7.4 focus on the extensions of *Mini-TACITUS* enabling the integration of reasoning with ontologies and similarity spaces into the abductive inference procedure.

### 7.1 Adapting *Mini-TACITUS* to a Large Knowledge Base

*Mini-TACITUS* (Mulkar *et al.*, 2007) began as a simple backchaining theorem-prover intended to be a more transparent version of the original *TACITUS* system, which was based on Stickel's *PTTP* system (Stickel, 1988). Originally, *Mini-TACITUS* was not designed for treating large amounts of data. A clear and clean reasoning procedure rather than efficiency was in the focus of its developers. For making the system work with a large knowledge base, several optimization steps had to be performed and a couple of new features had to be added.[1]

[1] We thank Niloofar Montazeri, Rutu Mulkar-Mehta, and Jerry Hobbs for their extensive help with the optimization of the *Mini-TACITUS* system.

***Time and Depth Parameters***

For controlling the reasoning complexity problem, two parameters were introduced.[2] The time parameter $t$ is used to restrict the processing time. If the processing time exceeds $t$, reasoning terminates and the best interpretation so far is output. The time parameter ensures that an interpretation will be always returned by the procedure even if reasoning could not be completed in a reasonable time.

The depth parameter $d$ restricts the depth of the inference chain. Suppose a proposition $p$ occurring in the input was backchained upon and a proposition $p'$ was introduced as the result. Then, $p'$ will be backchained upon and so on. The number of such iterations cannot exceed $d$. The depth parameter reduces the number of reasoning steps. Suppose the knowledge base contains the axioms $dog(e,x) \rightarrow mammal(e,x)$, $mammal(e,x) \rightarrow animal(e,x)$, $animal(e,x) \rightarrow organism(e,x)$ and the proposition $organism(e,x)$ occurs in the input logical form. Given the depth parameter value equal to 2, *organism* can be backchained on to *animal* (first step) and *mammal* (second step), but not to *dog*.

The interaction between the time and depth parameters is shown in Algorithm 7.1. A logical form $LF$, a knowledge base $KB$, a depth parameter $D$, a default cost parameter $C$, and a time parameter $T$ are input to the algorithm. Propositions from LF, with assigned initial default costs $C$ and depth equal to 0, constitute the initial interpretation *I_init* in the interpretation set *I_set*. Then, the recursive subroutine *apply_inference* is called, so that the initial interpretation *I_init* with assigned costs and depth is passed to the subroutine as a parameter.

As long as the processing time does not exceed $T$, the subroutine *apply_inference* works as follows. For each axiom $\alpha$ from the knowledge base $KB$, for each subset $PS$ of propositions belonging to the input interpretation $I$ such that these propositions have a depth smaller than the depth parameter $D$ and $\alpha$ can be applied to $PS$, the procedure constructs a new interpretation $I_{new}$ as the result of the application of $\alpha$ to $PS$. The depth of all propositions from $I$, which occur in $I_{new}$, is increased by 1. The new interpretation $I_{new}$ is added to the interpretation set *I_set*. The subroutine *apply_inference* is called again, with the parameter $I_{new}$.

After the subroutine *apply_inference* terminates, a set of interpretations having the lowest cost (*Cheapest_I*) is selected from the overall set *I_set* of constructed interpretations. Those interpretations from the *Cheapest_I* set, which have the shortest proof length (the

[2] The parameters were implemented into *Mini-TACITUS* by Rutu Mulkar-Mehta.

smallest sum of depths of all propositions), constitute the set of the best interpretations *Best_I*. The procedure returns the first element $I_{best}$ of this set.

Obviously, the time restriction can prevent the system from finding the "real" best interpretation, because there will be not enough time for applying all relevant axioms. As shown by experiments described in Sec. 8.3, this, indeed, happens when processing longer text fragments.

The depth restriction does not crucially influence the resulting best interpretation, because most of the axioms in the knowledge base presuppose not more than 3-4 backchaining steps. On the first backchaining step, a lexeme is mapped to a word sense (synset or frame). On the next steps, semantic relations defined on word senses are applied.

### *Filtering out Axioms and Input Propositions*

Since *Mini-TACITUS* processing time increases exponentially with the input size (sentence length and number of axioms), making such a large set of axioms work was an additional issue. For speeding up reasoning it was necessary to reduce both the number of the input propositions and the number of axioms. Those axioms and input propositions were filtered out, which could not contribute to the resulting inferences. Suppose that the initial logical form contains the following propositions:

$a(x_1,\ldots,x_n) \wedge b(y_1,\ldots,y_m) \wedge c(z_1,\ldots,z_k)$,

and the knowledge base consists of the following axioms:

(1) $d(x_1,\ldots,x_l) \rightarrow a(y_1,\ldots,y_n)$
(2) $g(x_1,\ldots,x_s) \wedge e(y_1,\ldots,y_r) \rightarrow d(z_1,\ldots,z_l)$
(3) $a(x_1,\ldots,x_n) \rightarrow b(y_1,\ldots,y_m) \wedge f(z_1,\ldots,z_t)$
(4) $g(x_1,\ldots,x_s) \rightarrow b(y_1,\ldots,y_m)$
(5) $a(x_1,\ldots,x_n) \rightarrow b(y_1,\ldots,y_m)$
(6) $h(x_1,\ldots,x_q) \rightarrow c(y_1,\ldots,y_k)$

First, only those axioms, which could be evoked by the input propositions or as a result of backchaining from the input, are relevant for reasoning. Given the logical form above, axiom (3) is obviously useless; it can never be evoked, because the predicate $f$ does not occur in the input and cannot be added through backchaining. Axiom (2) can be evoked only if the value of the depth parameter exceeds 1.

The corresponding filtering procedure is shown in Algorithm 7.2. Given a logical form *LF*, a knowledge base *KB*, and a depth parameter *D*, the algorithm selects axioms, which

**Algorithm 7.1** *Mini-TACITUS* reasoning algorithm: interaction of the time and depth parameters.

**Require:** a logical form $LF$ of a text fragment, a knowledge base $KB$,
a depth parameter $D$, a cost parameter $C$, a time parameter $T$

**Ensure:** the best interpretation $I_{best}$ of $LF$

$I_{init} := \{p(e, x_1, \ldots, x_n, C, 0) | p(e, x_1, \ldots, x_n) \in LF\}$

$I\_set := \{I_{init}\}$

$apply\_inference(I_{init})$

$Cheapest\_I := \{I | I \in I\_set \text{ and } \forall I' \in I\_set : cost(I) \leq cost(I')\}$

$Best\_I := \{I | I \in Cheapest\_I \text{ and }$
$\forall I' \in Cheapest\_I : proof\_length(I) \leq proof\_length(I')\}$

**return** $I_{best}$, which is the first element of $Best\_I$

**Subroutine** *apply_inference*

**Require:** interpretation $I$

**while** $processing\_time < T$ **do**
  **for** $\alpha \in KB$ **do**
    **for** $PS \subseteq I$ such that $\forall p(e, x_1, \ldots, x_n, c, d) \in PS : d < D$ **do**
      **if** $\alpha$ is applicable to $PS$ **then**
        $I_{new}$ := result of application of $\alpha$ to PS
        $I\_set := I\_set \cup \{I_{new}\}$
        $apply\_inference(I_{new})$
      **end if**
    **end for**
  **end for**
**end while**

can be evoked by the propositions in $LF$. *Pred* is a set of predicate names occurring in $LF$. The output set of axioms $KB_{LF}$ is initialized with an empty set. The algorithm calls the subroutine *collect_axioms* with the depth parameter equal to 0.

The subroutine *collect_axioms* works as follows. For each axiom $\alpha$ in $KB$, which is not contained in the $KB_{LF}$ set, it checks whether all predicate names occurring on the right hand side of $\alpha$ also occur in the predicate name set *Pred*. If this is the case, then the set *Pred* is extended by the predicate names occurring on the left hand side of $\alpha$ which simulates the reasoning process disregarding variable merging. The value of the depth variable is increased by 1. If it does not exceed the depth parameter $D$ then the subroutine *collect_axioms* is called again.

**Algorithm 7.2** Algorithm for selecting axioms, which can be evoked given a logical form.

**Require:** logical form $LF$ of a text fragment, knowledge base $KB$,
depth parameter $D$,

**Ensure:** set of axioms $KB_{LF}$

*Preds* is a set of predicate names occurring in $LF$
$KB_{LF} := \{\}$
*collect_axioms*(0)
**return** $KB_{LF}$

**Subroutine** *collect_axioms*

**Require:** *depth*

**for** $\alpha \in KB$ such that $\alpha \notin KB_{LF}$ **do**
    $PR$ is the set of predicate names occurring on the right hand side of $\alpha$
    **if** $PR \subseteq Pred$ **then**
        $KB_{LF} := KB_{LF} \cup \{\alpha\}$
        $Preds := Preds \cup PL$, where $PL$ is a set of predicate name occurring on the left hand side of $\alpha$
    **end if**
**end for**
$depth := depth + 1$
**if** $depth <= D$ **then**
    *collect_axioms*(*depth*)
**end if**

In addition to the axioms, which can never be evoked, those axioms are useless, which can never lead to any proposition merging. For example, axiom (1) in the example above can be evoked by the input proposition $a(x_1, \ldots, x_n)$ introducing the new predicate $d$, which allows the procedure to evoke axiom (2) and introducing $g$.[3] The predicate $g$ can be also

[3]The backchaining from $a$ to $d$ and from $d$ to $g$ is possible only if the depth parameter exceeds 1.

introduced by axiom (4) evoked by $b$. Thus, $a(x_1,\ldots,x_n)$ and $b(y_1,\ldots,y_m)$ occurring in the input can be potentially backchained upon introducing the same predicate $g$, which implies redundancy. Axiom (5) directly allows the procedure to backchain on $b$ introducing $a$. At the same time, axiom (6) introduces the predicate $h$, which otherwise never occurs in the input and cannot be introduced by any other axiom; therefore, it cannot be merged with anything else. Thus, axiom (6) is useless for abductive reasoning.

Similarly, proposition $c(z_1,\ldots,z_k)$ in the input logical form can never be merged with any other proposition and can never evoke an axiom introducing a proposition, which can be merged with any other. Therefore, removing the proposition $c(z_1,\ldots,z_k)$ from the input for the reasoning machine and adding it to the best interpretation after the reasoning terminates (replacing its arguments with new variables if mergings took place) does not influence the reasoning process.

In logical forms, propositions, which could not be linked to the rest of the discourse, often refer to modifiers. For example, consider the sentence *Yesterday, John bought a book, but he has not started reading it yet*. The information concerning John buying a book is in the focus of this text fragment; it is linked to the second part of the sentence. However, the modifier *yesterday* just places the situation in time; it is not connected to any other part of the discourse.

The algorithm for filtering out axioms and propositions, which can never lead to any merging, is based on a backchaining matrix $BM$. $BM$ is a quadratic matrix with as many rows as there are predicate names in the knowledge base. If a predicate $q$ can be backchained upon to $p$ using an axiom from KB, then the cell $BM[q,p]$ contains a set including the index of the corresponding axiom; otherwise it contains an empty set. The backchaining matrix constructed for the example above has the following form[4]:

| | **a** | **b** | **c** | d | e | g | h |
|---|---|---|---|---|---|---|---|
| **a** | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\{1\}$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |
| **b** | $\{5\}$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\{4\}$ | $\varnothing$ |
| **c** | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\{6\}$ |
| d | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\{2\}$ | $\{2\}$ | $\varnothing$ |
| e | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |
| g | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |
| h | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |

[4]Axiom (3) is excluded from the consideration by Algorithm 7.2.

This matrix is iteratively extended with transitive inference chains as follows: If $BM[x,y] = S_1$ and $BM[y,z] = S_2$ then the value of the cell $BM[x,z]$ becomes $BM[x,z] \cup S_1 \cup S_2$. The number of iterations equals the value of the depth parameter. Given the depth parameter value 1, the matrix above will be changed to the following:

| | **a** | **b** | **c** | d | e | g | h |
|---|---|---|---|---|---|---|---|
| **a** | $\varnothing$ | $\varnothing$ | $\varnothing$ | {1} | {1,2} | {1,2} | $\varnothing$ |
| **b** | {5} | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | {4} | $\varnothing$ |
| **c** | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | {6} |
| d | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | {2} | {2} | $\varnothing$ |
| e | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |
| g | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |
| h | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |

Given the extended backchaining matrix, we want to know, which axioms can help to recover implicit redundancy leading to the reduction of the interpretation cost. In other words, we want to know which axioms enable backchaining on two different propositions from the logical form, so that the same predicate is introduced several times. In our example, backchaining on $a$ and $b$ results in introducing the same predicate twice. First, $a$ and $b$ both introduce $a$ if axiom (5) is applied to $b(y_1,\ldots,y_m)$. Second, both $a$ and $b$ can be backchained on to $g$ using axioms (1), (2), and (4). Thus, axioms (1), (2), (4), and (5) are potentially useful for interpreting the input logical form, while axioms (3) and (6) can be excluded from the consideration.

Given the backchaining matrix $BM$, the set of "useful" axioms $UA$ can be defined as follows: $UA = \{\alpha \in AI \mid \exists x,y \in Preds : \alpha \in BM[x,y] \textit{ or } \exists\ z : \alpha \in BM[x,z] \cup BM[y,z]\}$, where $AI$ is a set of axiom indices and $Preds$ is a set of predicate names occurring in the input logical form.

Similarly, the set of "useful" predicate names $UP$ of the input propositions is defined as follows: $UP = \{x \in Preds \mid \exists y \in Preds : BM[x,y] \neq \varnothing \textit{ or } BM[y,x] \neq \varnothing \textit{ or } (\exists z \in Preds : BM[x,z] \neq \varnothing \textit{ and } BM[y,z] \neq \varnothing)\}$, where $Preds$ is a set of predicate names occurring in the input logical form. Propositions with predicate names not belonging to the "useful" set can be excluded from the reasoning process and added to the best interpretation after application of axioms has been finished. Then, the substitutions of variables resulting from the factorization process should be applied to these propositions.[5] In the example above, such a "useless" proposition is proposition $c(z_1,\ldots,z_k)$.

In the experiments described in Sec. 8.3, filtering out axioms applying Algorithm 7.2 resulted in a reduction of axiom set size by a factor of 4 per sentence on average. Reducing the number of input propositions resulted in filtering out one third of the input propositions per sentence on average.

## 7.2 Refining Abductive Reasoning Procedure for NLU

In practical application of weighted abduction to NLU tasks, we encountered several pragmatic problems discussed in this section. First, we introduce constraints preventing undesired mergings of propositions having the same predicate name. Second, we concerns the problem of circular definitions.

### *Non-Merge Constraints*

Frequently, the lowest-cost interpretation in weighted abduction results from identifying two entities with each other, so that their common properties only need to be proved or assumed once. This feature of the algorithm is one of the principal methods by which coreference is resolved. The abductive reasoning procedure as described in Sec. 4.3 performs merging of propositions having the same predicate names at every reasoning step. Consider the text fragment *Pluto is a dog. This animal is funny* with the following logical form:

$Pluto(e_1,x_1) \wedge dog(e_2,x_1) \wedge animal(e_3,x_2) \wedge funny(e_4,x_2)$

If the knowledge base contains the axiom $dog(e_1,x_1) \rightarrow animal(e_1,x_1)$, then the logical form above will be expanded into the following:

$Pluto(e_1,x_1) \wedge dog(e_2,x_1) \wedge animal(e_3,x_2) \wedge funny(e_4,x_2) \wedge dog(e_3,x_2)$

Since the propositions $dog(e_2,x_1)$ and $dog(e_3,x_2)$ have the same predicate name *dog*, they will be merged so that $x_1$ will be set equal to $x_2$ resulting in the following logical form:

$Pluto(e_1,x_1) \wedge dog(e_2,x_1) \wedge animal(e_2,x_1) \wedge funny(e_4,x_1)$

As mentioned in Sec. 4.5, merging propositions with the same predicate names does not always give the intended solution. For example, in the sentence *John eats an apple and Bill*

[5]We thank Niloofar Montazeri for the help with implementation of the proposition filtering option into *Mini-TACITUS*.

*eats an apple* the two *eat* propositions should not be merged because they refer to different actions. Similarly, two articles *a* in the sentence *A dog eats a bone* should not be merged, because such merging would imply that both articles and the corresponding nouns refer to the same entity, which is obviously wrong. The same problem occurs with the preposition *of* in the text fragment *handling of conflicts of interest* and with many other predicates.

The *Mini-TACITUS* system allows us to define non-merge constraints, which prevent undesirable mergings at every reasoning step. Non-merge constraints have the form of $x_1 \neq y_1, \ldots, x_n \neq y_n$. These constraints are generated by the system at each reasoning step. Given the propositions $p(x_1)$ and $p(x_2)$ occurring in the input logical form and the non-merge constraint $x_1 \neq x_2$, *Mini-TACITUS* does not merge $p(x_1)$ and $p(x_2)$, because it would imply a conflict with the non-merge constraint.

In the experiments described in this book, we used the following rule for generating non-merge constraints[6]:

*For each two propositions $p(e_1, x_1, \ldots, x_n)$ and $p(e_2, y_1, \ldots, y_n)$, which occur in the input, if*

- *$e_1$ is not equal to $e_2$,*
- *p is not a noun predicate, and*
- *$\exists i \in \{1, \ldots, n\}$ such that $x_i$ is not equal to $y_i$ and both, $x_i$ and $y_i$, occur as arguments of any other proposition than p,*

*then add $e_1 \neq e_2$ to the non-merge constraints.*

This rule ensures that nouns can be merged without any restriction and other predicates can be merged only if all their non-first arguments are equal (due to the previous mergings) or uninstantiated. As seen from the statements above, the argument merging restriction concerns first arguments only. First arguments of all predicates in the logical forms treated by *Mini-TACITUS* are “handles” referring to conditions, in which the predicate is true of its arguments, i.e. referring to the predication itself, rather than to its semantic arguments.

The proposed non-merge rule is a heuristic, which corresponds to the intuition that it is unlikely that the same noun refers to different entities in a short discourse, while for other predicates this is possible. According to this rule the two *eat* propositions can be merged

[6]We thank Niloofar Montazeri for the help with implementation of the non-merge constraints into *Mini-TACITUS*.

in the sentence *John eats an apple and he eats the fruit slowly* having the following logical form[7]:

$$John(e_1,x_1) \wedge eat(e_2,x_1,x_2) \wedge apple(e_3,x_2) \wedge and(e_4,e_2,e_5)$$
$$he(e_1,x_1) \wedge eat(e_5,x_1,x_3) \wedge fruit(e_6,x_3) \wedge slowly(e_7,e_5)$$

In the logical form above, the propositions $eat(e_2,x_1,x_2)$ and $eat(e_5,x_1,x_3)$ cannot be merged, because they do not refer to nouns and their third arguments $x_2$ and $x_3$ are not equal. If the knowledge base contains the axiom $apple(e_1,x_1) \rightarrow fruit(e_1,x_1)$ then the logical form above can be expanded into the following:

$$John(e_1,x_1) \wedge eat(e_2,x_1,x_2) \wedge apple(e_3,x_2) \wedge and(e_4,e_2,e_5)$$
$$he(e_1,x_1) \wedge eat(e_5,x_1,x_3) \wedge fruit(e_6,x_3) \wedge apple(e_6,x_3) \wedge slowly(e_7,e_5)$$

After the expansion, the noun propositions $apple(e_3,x_2)$ and $apple(e_6,x_3)$ can be merged resulting in the following logical form:

$$John(e_1,x_1) \wedge eat(e_2,x_1,x_2) \wedge apple(e_3,x_2) \wedge and(e_4,e_2,e_5)$$
$$he(e_1,x_1) \wedge eat(e_5,x_1,x_2) \wedge fruit(e_3,x_2) \wedge slowly(e_7,e_5)$$

Now, when all the arguments of the two *eat* propositions are equal, these propositions can be merged as well, which results in the following logical form:

$$John(e_1,x_1) \wedge eat(e_2,x_1,x_2) \wedge apple(e_3,x_2) \wedge and(e_4,e_2,e_5)$$
$$he(e_1,x_1) \wedge fruit(e_6,x_2) \wedge slowly(e_7,e_5)$$

Concerning the sentence *John eats an apple and Bill eats an apple*, merging of two *read* propositions is impossible, unless the system manages to prove that the predicates *John* and *Bill* can refer to the same individual.

There are cases when the proposed rule does not block undesired mergings. For example, given the sentence *John owns red apples and green apples*, it is wrong to merge both *apple* propositions, because "being red" and "being green" are incompatible properties that cannot be both assigned to the same entity. Thus, it seems to be reasonable to check, whether two propositions to be merged have incompatible properties. An additional study of reference resulting in additional non-merge rules is needed to address such cases.

### *Circular Axioms*

In the abductive framework, circular axioms in the knowledge base do not cause infinite inference loops, but they can result in assigning inadequate proposition costs. Suppose the knowledge base contains the axioms (1) $P^{w_1} \rightarrow Q$ and (2) $Q^{w_2} \rightarrow P$, and the proposition

[7]The anaphoric *he* in the logical form is already linked to its antecedent *John*.

$Q$ with the cost $c$ occurs in the input. After applying axiom (1), the input formula will be expanded into $Q : 0 \wedge P : w_1 \cdot c$, where $Q$ has the cost of 0 and $P$ costs $w_1 \cdot c$. Now, axiom (2) can be applied resulting in the following formula: $Q : 0 \wedge P : 0 \wedge Q : w_1 \cdot c \cdot w_2$. The two $Q$ propositions will be merged, so that the final interpretations looks as follows: $P : 0 \wedge Q : 0$. This means that the input formula $b$ was fully proven, which is obviously wrong.

In practice, circular axioms are generated on the basis of an equality relation. For example, the FrameNet relation *Perspective* (see Sec. 5.2.2) is converted into a bidirectional implication: NEUTRAL_FRAME → PERSPECTIVE_FRAME, PERSPECTIVE_FRAME → NEUTRAL_FRAME.

In order to avoid circular reasoning, the following feature was added to *Mini-TACITUS*. For each proposition $p$ in the logical form, the systems keeps information about the propositions, which were backchained on in order to enable the assumption of $p$. For example, given the axiom $A \wedge B \rightarrow C \wedge D$ and the input formula $C \wedge D$ expanded into $A \wedge B \wedge C \wedge D$, the system will know that $A$ and $B$ were introduced due to $C$ and $D$.

Then, for each axiom $P_1 \wedge \ldots \wedge P_m \rightarrow Q_1 \wedge \ldots \wedge Q_n$ applicable to the input, the system checks whether some $Q_j$ $(1 \leq j \leq n)$ in the input was introduced due to some $P_i$ $(1 \leq i \leq m)$. If this is the case, then the axiom application is blocked.

The proposed approach allows us to keep circular definitions in the knowledge base, but it blocks circular inferences.[8] The drawback of this approach concerns the need of tracking "parent" propositions for each introduced assumption, which increases the memory space required for storing each constructed interpretation.

Given the described extensions, on each reasoning step the abductive reasoning procedure

(1) generates non-merge constraints,
(2) merges propositions with the same predicate name, assigning the lowest cost to the result of merging, if it does not contradict the non-merge constraints,
(3) applies an axiom to a conjunction of propositions, which has a non-zero cost if this does not introduce circles.

The corresponding full abductive algorithm is presented in Appendix. As already mentioned, the processing time increases exponentially with the input size (sentence length and number of axioms). The algorithm terminates, because the backchaining steps are restricted

[8] We thank Niloofar Montazeri for implementation of the blocking of circular reasoning into *Mini-TACITUS*.

by the depth parameter. The correctness of the algorithm can not be proven formally, because abductive inference is not a valid inference.

## 7.3 Reasoning with Ontologies

As mentioned in Sec. 5.3, the proposed integrative knowledge base stores ontologies in a separate module and queries them using a Description Logic (DL) reasoner. Reasoning with ontologies given an input logical form proceeds as follows.

First, mappings from parts of the logical form into ontological concept descriptions are applied. Consider the following text fragment:

*The company is actively selling cars with 3 doors.*

*Fiat Punto Grande is especially popular.*

with the following logical form[9]:

$$company(e_1,x_1) \wedge sell(e_2,x_1,x_2) \wedge car(e_3,x_2) \wedge with(e_4,x_2,x_3) \wedge$$
$$3(e_5,x_3) \wedge door(e_6,x_3) \wedge fiat(e_7,x_4) \wedge punto\text{-}grande(e_8,x_5) \wedge$$
$$nn(e_9,x_4,x_5) \wedge popular(e_{10},x_4)$$

Suppose that there are the following lexicon-ontology mappings[10]:

(1) $\text{FIAT\_PUNTO\_GRANDE}(x_1) \rightsquigarrow$
$fiat(e_1,x_1) \wedge punto\text{-}grande(e_2,x_2) \wedge nn(e_3,x_1,x_2)$

(2) $\text{CAR}(x_1) \rightsquigarrow car(e_1,x_1)$

(3) $(\leq 3\ \text{HAS\_DOORS} \sqcap \geq 3\ \text{HAS\_DOORS})(x_1) \wedge \text{HAS\_DOORS}\ (x_1,x_2) \rightsquigarrow$
$with(e_1,x_1,x_2) \wedge 3(e_2,x_2) \wedge door(e_3,x_2)$

Given the mappings above, the initial logical form can be extended as follows:

$$company(e_1,x_1) \wedge sell(e_2,x_1,x_2) \wedge car(e_3,x_2) \wedge with(e_4,x_2,x_3) \wedge$$
$$3(e_5,x_3) \wedge door(e_6,x_3) \wedge \text{CAR}(x_2) \wedge (\leq 3\text{HAS\_DOORS} \sqcap \geq 3\text{HAS\_DOORS})(x_2) \wedge$$
$$\text{HAS\_DOORS}(x_2,x_3) \wedge fiat(e_7,x_4) \wedge punto\text{-}grande(e_8,x_5) \wedge nn(e_9,x_4,x_5) \wedge$$
$$\text{FIAT\_PUNTO\_GRANDE}(x_4) \wedge popular(e_{10},x_4)$$

[9]For the sake of simplicity, propositions irrelevant for this example are omitted in the logical form.
[10]Recall that these mappings are not logical axioms (Sec. 5.3). Therefore the symbol $\rightsquigarrow$ instead of the implication symbol $\rightarrow$ is used here. The DL concept descriptions (e.g., $\leq 3$ HAS_DOORS $\sqcap \geq 3$ HAS_DOORS) are treated by the abductive reasoner as strings of symbols. They are used as logical formulas only when a Description Logic reasoner is employed.

Given the logical form extended with ontological concepts, an ontological assertion box (see Sec. 4.4) can be constructed. For the example above, the following assertion box $\mathscr{A}_1$ is constructed:

$$\mathscr{A}_1 = \{(\textsc{Car} \sqcap \leq 3\textsc{has\_Doors} \sqcap \geq 3\textsc{has\_Doors})(x_2),$$
$$\textsc{has\_Doors}(x_2, x_3), \textsc{Fiat\_Punto\_Grande}(x_4)\}$$

Suppose that the DL terminological box $\mathscr{T}_1$ of the corresponding ontology contains the following two axioms:

$$\mathscr{T}_1 = \{\textsc{Fiat\_Punto\_Grande} \sqsubseteq \textsc{Fiat} \sqcap \leq 3\ \textsc{has\_Doors} \sqcap \geq 3\ \textsc{has\_Doors},$$
$$\textsc{Fiat} \sqsubseteq \textsc{Car}\ \}$$

Given the terminological box $\mathscr{T}_1$ and the assertion box $\mathscr{A}_1$ above, it is desirable to infer $\textsc{Fiat}(x_4)$ and to set $x_2$ and $x_4$ to be equal. These inferences can be obtained by employing a model builder, similar to the approach by Bos (2003) and Cimiano (2003) (see Sec. 4.2 for more details). The model builder is supposed to construct a minimal model $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$ of an ABox $\mathscr{A}$ towards a TBox $\mathscr{T}$ such that for any other model $\mathscr{I}' = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$ of $\mathscr{A}$: $|\Delta^{\mathscr{I}}| \leq |\Delta^{\mathscr{I}'}|$, i.e. $\mathscr{I}$ introduces less individuals than any other model.

In a minimal model, all individuals are set to be equal, if it does not introduce inconsistencies (see Sec. 4.2). In order to use such models, one needs to be sure that the ontology contains disjointness axioms for all incompatible concepts, e.g., $\textsc{Car} \sqsubseteq \neg\textsc{Door}$ and $\textsc{Door} \sqsubseteq \neg\textsc{Car}$. However, such axioms dramatically increase the size of the ontology. Moreover, most of the ontology engineers do not care about introducing all possible disjointness axioms. Therefore, instead of merging as much individuals as possible in a minimal model, we propose to merge only those individuals, which instantiate the same concept descriptions and relations, similar to the approach of weighted abduction (see Sec. 4.3). For constructing such a merging model, expansion trees produced by the tableau algorithm described in Sec. 4.4 can be reduced as follows.

*If an expanded ABox $\mathscr{A}$ constructed towards a TBox $\mathscr{T}$*

- *contains $C(x_1), C(x_2)$ and*
- *$C_1 \sqcap ... \sqcap C_n$ is satisfiable towards $\mathscr{T}$, such that $\forall i \in \{1,\dots,n\} : C_i(x_1) \in \mathscr{A}$ and $C_i(x_2) \in \mathscr{A}$*[11]*,*

*then introduce a new variable $d$ and replace $x_1$ and $x_2$ with $d$ in $\mathscr{A}$.*

*Similarly, if*

[11]The model reducing procedure checks whether setting $x_1$ equal to $x_2$ introduces a clash in $\mathscr{A}$ (cf. Sec. 6.2 for the definition of clash).

- *$R(x_1,y_1), R(x_2,y_2) \in \mathscr{A}$ and*
- *$C_1 \sqcap \ldots \sqcap C_n$ is satisfiable towards $\mathscr{T}$, such that $\forall i \in \{1,\ldots,n\} : C_i(x_1) \in \mathscr{A}$ and $C_i(x_2) \in \mathscr{A}$ and*
- *$C_1 \sqcap \ldots \sqcap C_n$ is satisfiable towards $\mathscr{T}$, such that $\forall i \in \{1,\ldots,n\} : C_i(y_1) \in \mathscr{A}$ and $C_i(y_2) \in \mathscr{A}$ and*
- *$\forall i \in \{1,2\}$: if $(\geq nR)(x_i) \in \mathscr{A}$ then $|\{z : R(x_i,z) \in \mathscr{A}\}| \geq n+1$,*

*then introduce new variables $d_1$, $d_2$, replace $x_1$, $x_2$ with $d_1$, and replace $y_1$, $y_2$ with $d_2$.*

According to the rules above, individuals instantiating the same concept descriptions and relations can be set to be equal, if such mergings do not introduce inconsistencies. For relations, it should be also checked that corresponding at-least restrictions are not violated by the performed mergings.

If an input concept description evoked by the logical form contains disjunction or at-most restrictions, then more than one model for this concept description can be constructed (see Sec. 4.4). If, after being reduced, one model appears to be smaller than the others, then this model should be selected as the merging model. Otherwise, the first model having minimum size should be selected.

The ABox $\mathscr{A}_1$ expanded using the tableau algorithm introduced in Sec. 4.4 includes the following elements:

$$
\begin{aligned}
\mathscr{A}_1' = \{ & (\textsc{Car} \sqcap \leq 3\textsc{has\_Doors} \sqcap \geq 3\textsc{has\_Doors})(x_2),\\
& \textsc{has\_Doors}(x_2,x_3),\\
& \textsc{Car}(x_2), (\leq 3\textsc{has\_Doors} \sqcap \geq 3\textsc{has\_Doors})(x_2),\\
& \leq 3\textsc{has\_Doors}(x_2), \geq 3\textsc{has\_Doors}(x_2),\\
& \textsc{has\_Doors}(x_2,a_1), \textsc{has\_Doors}(x_2,a_2),\\
& \textsc{Fiat\_Punto\_Grande}(x_4), \textsc{Fiat}(x_4), \textsc{Car}(x_4),\\
& (\leq 3\textsc{has\_Doors} \sqcap \geq 3\textsc{has\_Doors})(x_4),\\
& \leq 3\textsc{has\_Doors}(x_4), \geq 3\textsc{has\_Doors}(x_4),\\
& \textsc{has\_Doors}(x_4,b_1), \textsc{has\_Doors}(x_4,b_2), \textsc{has\_Doors}(x_4,b_3)\}
\end{aligned}
$$

The elements $\textsc{Car}(x_2)$ and $\textsc{Car}(x_4)$ in $\mathscr{A}_1$ can be merged, because the concept description $\leq 3\textsc{has\_Doors} \sqcap \geq 3\textsc{has\_Doors} \sqcap \textsc{Fiat\_Punto\_Grande} \sqcap \textsc{Fiat}$ is satisfiable towards $\mathscr{T}_1$. The instantiations of the $\textsc{has\_Doors}$ relation can be merged as well. Thus, the merging model $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$ of $\mathscr{A}_1'$ is as follows[12]:

[12]For the sake of simplicity, complex concept descriptions containing conjunction are omitted in this model.

$\Delta^{\mathscr{I}} = \{d_1, d_2, d_3, d_4\}$,
$x_2 = d_1,\ x_4 = d_1,\ x_3 = d_2,\ a_1 = d_3,\ a_2 = d_4,\ b_1 = d_2,\ b_2 = d_3,\ b_3 = d_4$,
$\text{CAR}^{\mathscr{I}} = \{d_1\}$,
$(\leq 3\text{HAS\_DOORS})^{\mathscr{I}} = \{d_1\}$,
$(\geq 3\text{HAS\_DOORS})^{\mathscr{I}} = \{d_1\}$,
$(\text{HAS\_DOORS})^{\mathscr{I}} = \{(d_1, d_2), (d_1, d_3), (d_1, d_4)\}$
$\text{FIAT\_PUNTO\_GRANDE}^{\mathscr{I}} = \{d_1\}$,
$\text{FIAT}^{\mathscr{I}} = \{d_1\}$

This model can be used to enrich the original logical form as follows:

$company(e_1, x_1) \wedge sell(e_2, x_1, x_2) \wedge car(e_3, x_2) \wedge with(e_4, x_2, x_3) \wedge$
$3(e_5, x_3) \wedge door(e_6, x_3) \wedge \text{CAR}(x_2) \wedge (\leq 3\text{HAS\_DOORS} \sqcap \geq 3\text{HAS\_DOORS})(x_2) \wedge$
$\text{HAS\_DOORS}(x_2, x_3) \wedge \text{HAS\_DOORS}(x_2, x_6) \wedge \text{HAS\_DOORS}(x_2, x_7) \wedge$
$fiat(e_7, x_2) \wedge punto\text{-}grande(e_8, x_5) \wedge nn(e_9, x_2, x_5) \wedge$
$\text{FIAT\_PUNTO\_GRANDE}(x_2) \wedge popular(e_{10}, x_2) \wedge \text{FIAT}(x_2)$

In this logical form, the propositions $car(x_2)$ and $fiat(x_2)$ refer to the same entity; moreover, new propositions corresponding to the ontological concept FIAT and relation HAS_DOORS are added. The extended logical form can be further processed by *Mini-TACITUS* using the abductive knowledge base. In this case, no further merging is performed on propositions having ontological concept descriptions as predicate names.

Given a terminological box $\mathscr{T}$ and a logical form, the proposed steps for reasoning with ontologies within an abductive NLU pipeline can be summarized as follows.

(1) Apply lexicon-ontology mappings to map parts of the logical form into ontological concept descriptions.
(2) Construct an ABox $\mathscr{A}$ using the ontological concept descriptions evoked by the logical form.
(3) Apply a tableau algorithm for constructing a model/models for $\mathscr{A}$ towards $\mathscr{T}$.
(4) Reduce the model/models by merging instantiations of the same concept.
(5) Chose a merging model.
(6) Add the merging model to the logical form changing the variables according to the mergings in the model.

Unfortunately, the existing DL reasoners do not output constructed models, even if they are using a tableau algorithm. An additional implementation is needed to access the models. One could employ a more general deductive model builder (see Sec. 4.2) instead of a DL reasoner. However, since DL reasoning machines usually require less processing time and space than FOL theorem-provers (see Baader *et al.*, 2003), and most of the existing domain-specific ontologies are formalized in a DL-based format, the usage of DL reasoners is preferred. We introduce a simplified version of an NLU reasoning procedure employing ontologies. Given a terminological box $\mathscr{T}$ and a logical form, the procedure consists in the following steps.

(1) Apply lexicon-ontology mappings to map parts of the logical form into ontological concept descriptions.
(2) Construct an ABox $\mathscr{A}$ using the ontological concept descriptions evoked by the logical form.
(3) If a) a concept description $C_1$ is subsumed by a concept description $C_2$ ($C_1 \sqsubseteq C_2$) towards the TBox $\mathscr{T}$ and b) $C_1(x_1), C_2(x_2) \in \mathscr{A}$ and c) for all $D_1(x_1), D_2(x_2) \in \mathscr{A}$, $D_1 \sqcap D_2$ is satisfiable towards $\mathscr{T}$, then
   - set the costs of the propositions in the logical form, which evoke $C_2(x_2)$, equal to 0 and
   - set $x_1$ be equal to $x_2$ in the logical form.

The idea of this simplified procedure is the following. If a concept description $C_2$ subsumes a more specific concept description $C_1$, then $C_1$ entails $C_2$ and, given the concept $C_2$, the concept $C_1$ can be considered to be proven. Moreover, the corresponding variables can be set to be equal, because $C_1 \sqsubseteq C_2$ implies that $C_1$ and $C_2$ refer to the same entity.

Concerning the logical form under consideration, since $(\textsc{Car} \sqcap \leq 3\textsc{has\_Doors} \sqcap \geq 3\textsc{has\_Doors})$ subsumes $\textsc{Fiat\_Punto\_Grande}$ towards $\mathscr{T}_1$, the costs of the propositions $car(e_3,x_2) \wedge with(e_4,x_2,x_3) \wedge 3(e_5,x_3) \wedge door(e_6,x_3)$ are set equal to 0 and the variables $x_2$ (argument of *car*) and $x_4$ (argument of *fiat*) are set to be equal, so that the resulting logical form looks as follows.

$$company(e_1,x_1) \wedge sell(e_2,x_1,x_2) \wedge car(e_3,x_2) \wedge with(e_4,x_2,x_3) \wedge$$
$$3(e_5,x_3) \wedge door(e_6,x_3) \wedge \textsc{Car}(x_2) \wedge (\leq 3\textsc{has\_Doors} \sqcap \geq 3\textsc{has\_Doors})(x_2) \wedge$$
$$\textsc{has\_Doors}(x_2,x_3) \wedge fiat(e_7,x_2) \wedge punto\text{-}grande(e_8,x_5) \wedge nn(e_9,x_2,x_5) \wedge$$
$$\textsc{Fiat\_Punto\_Grande}(x_2) \wedge popular(e_{10},x_2)$$

This logical form does not contain additional concepts and relations ($\textsc{Fiat}$, $\textsc{has\_Doors}$), which can be introduced by a DL model, but it still properly links $car(x_2)$

to $fiat(x_4)$. It is easy to implement, because it does not require any additional development of the existing reasoners. Since this study is not focused on implementation of a new reasoner, but rather relies on the existing tools, we employed this simplified procedure for reasoning with ontologies in the experiment described in Sec. 8.3.4. In this experiment, the *HermiT*[13] reasoner for Description Logics was used (see Sec. 4.4).

The proposed approach can be compared to the approach to abductive reasoning with DL ontologies for discourse interpretation as introduced by Peraldi *et al.* (2008) (for technical details, see Castano *et al.*, 2008). Peraldi *et al.* (2008) use deductive model building with integrated abductive backchaining, which is based on an additional knowledge base consisting of non-recursive flat conjunctive rules similar to the axioms in (Hobbs *et al.*, 1993). The advantage of the proposed approach as compared to the one presented in (Peraldi *et al.*, 2008) is that it fully relies on standard DL reasoning service and does not require an extension of the TBox with additional rules.

## 7.4 Reasoning with Similarity Space

As mentioned in Sec. 5.4, we propose to use semantic similarity in order to detect relationships between those parts of the discourse, which cannot be related with the help of the axioms. Thus, reasoning with similarity space is carried out after the application of axioms from the knowledge base has been finished. For those propositions, which have non-zero costs, i.e. those propositions, which have not been backchained on, information about their similarity to the rest of the discourse is used to reduce their costs.

Semantic similarity is computed using a similarity vector space (see Sec. 5.4 for more details). Given the similarity space, term similarity is determined by measuring the cosine of the angle between the corresponding term vectors; negative values are set to 0; similarity values are normalized so that they range from 0 to 1.

Consider the following text fragment:

*Barack Obama has visited Germany. The U.S. president has been to Berlin.*

The anaphora in the logical form below, which corresponds to this sentence is unresolved.

$$Barack\text{-}Obama(e_1,x_1) \wedge visit(e_2,x_1,x_2) \wedge Germany(e_3,x_2) \wedge$$
$$U.S.(e_4,x_3) \wedge president(e_5,x_4) \wedge nn(e_6,x_3,x_4) \wedge be(e_7,x_4) \wedge$$
$$to(e_8,e_7,x_5) \wedge Berlin(e_9,x_5)$$

[13] *http://www.comlab.ox.ac.uk/projects/HermiT/index.html*

Suppose that the knowledge base contains no axioms about Barack Obama and the similarity space provides the following measures:

| | | |
|---|---|---|
| *Barack-Obama* | *U.S.* | 0.9 |
| *Barack-Obama* | *president* | 0.9 |
| *Barack-Obama* | *Germany* | 0.2 |
| *Barack-Obama* | *Berlin* | 0.1 |
| *Barack-Obama* | *visit* | 0.5 |

Since the information about similarity implies no argument structure, it cannot be used for merging propositions. However, this information can be used to reduce costs of those propositions, which cannot be linked to the rest of the discourse with the help of axioms, but, which are nevertheless related to it. In the example above, the term *Barack-Obama* is strongly related to the terms $U.S.$ and $president$. Two different strategies can be applied to exploit this knowledge:

(1) For each proposition $P$ in the logical form, an average similarity $s_a$ between its predicate name and the predicate names of all other propositions in the logical form is computed. The cost of $P$ will be then reduced to $(1 - s_a) \cdot cost(P)$.

(2) For each proposition $P$ in the logical form, similarity $s_i$ between its predicate name and the predicate names of all other propositions $P_i$ in the logical form is computed. Let $max_s$ be the maximum of $s_i$ over all $i$. The cost of $P$ will be then reduced to $(1 - max_s) \cdot cost(P)$.

The first option above presupposes that we measure in how far $P$ is entailed by the whole context. Taking the second option, we try to find at least one strong connection between $p$ and some other proposition in the logical form. Concerning the example above, the first option implies that the cost of *Barack-Obama*$(e_1,x_1)$ should be reduced (1-(0.9+0.9+0.2+0.1+0.5)/5)=0.49 times. If we follow the second option then the cost of *Barack-Obama*$(e_1,x_1)$ should be reduced (1-*max*(0.9+0.9+0.2+0.1+0.5))=0.1 times.

Obviously, using semantic similarity to reduce proposition costs does not help to infer implicit relationships as it can be done with the help of structured axioms. For example, it does not help to detect that *Barack Obama* and *U.S. president* in the example above refer to the same individual. However, using semantic similarity can help to obtain a better scaled cost of the best interpretation, which can be seen as a measure of text coherence. Moreover, it is useful for applications implying matching of two text fragments like, for example, recognizing textual entailment (RTE). Section 8.3.1 investigates the impact of semantic similarity in the RTE task.

## 7.5 Concluding Remarks

In this chapter, we described extensions and optimization steps performed on the abductive reasoning system *Mini-TACITUS* in order to make the system able to work with the proposed integrative knowledge base.

Optimization steps are described in Sec. 7.1. Two parameters were introduced, which restrict processing time and backchaining depth. After the processing time exceeds the value of the time parameter, the reasoning terminates and the best interpretation so far is output. This ensures that an interpretation will be always returned by the system even if reasoning could not be completed in reasonable time. Obviously, time restrictions can prevent the system from applying all relevant axioms and finding the "real" best interpretation, as shown by the experiments described in Sec. 8.3.

Since *Mini-TACITUS* processing time increases exponentially with the input size, reducing the number of the input propositions and the number of axioms results in speeding up reasoning. The developed procedure for reducing the size of the input filters out those axioms and input positions, which are useless for reasoning, because they can never reveal redundancy and lead to a proposition merging. This optimization resulted in a significant reduction of the size of the input propositions and axioms used for reasoning.

The performed optimization steps allow us to employ the integrative knowledge base in the experiments described in Chap. 8. However, as the results of the experiments suggest, there is still a need for speeding up the reasoner.

Section 7.2 is devoted to two pragmatic problems encountered in practical application of weighted abduction to NLU. First, constraints preventing undesirable mergings of propositions are introduced. As mentioned in the corresponding subsection, the proposed rule for generating non-merge constraints does not cover all problematic cases. An additional study of reference is required to improve on it. Second, a solution to the problem of circular definitions is discussed. The proposed solution is based on tracking propositions responsible for each backchaining step.

Section 7.3 focuses on reasoning with ontologies embedded in the abductive reasoning pipeline. In the developed integrative knowledge base, ontologies are stored in a separate module and queried by a specialized DL reasoning system. One possibility for integrating ontological reasoning into the abductive reasoning pipeline concerns the usage of DL models. Unfortunately, existing DL reasoners do not output constructed models, and an additional implementation is required to access the models. Therefore, we propose a simplified procedure for reasoning with ontologies, which implies merging two propositions if they evoke concept descriptions $C_1$ and $C_2$ such that $C_1$ is subsumed by $C_2$ towards an ontology. This procedure fully relies on the inference services provided by existing DL reasoners. However, it does not enable us to infer new relationships between concepts, which are implied by the ontology. Therefore, future work includes extending an existing DL reasoning machine for making it able to output models.

Section 7.4 describes how a similarity space can be employed in order to link those propositions, for which there are no axioms in the knowledge base to the rest of the discourse. For those propositions, which have not being backchained on during abductive reasoning, information about their similarity to the rest of the discourse is used to reduce their costs. Although using semantic similarity to reduce proposition costs does not help to infer implicit relationships as it can be done with the help of structured axioms, it can help to obtain more realistic interpretation costs and solve practical tasks implying matching of two text fragments like, for example, recognizing textual entailment (see Sec. 8.3).

Figure 7.1 shows the overall reasoning pipeline. First, the ontological reasoning is applied. An input logical form and a lexicon-ontology mapping are used by a mapping module to produce a DL ABox. This ABox together with an OWL ontology is processed by the DL reasoner *Hermit*. The reasoner checks, whether some two variables $x_1$ and $x_2$ instantiate concept descriptions $C_1$ and $C_2$ ($C_1(v_1)$ and $C_2(v_2)$ occur in the ABox) such that $C_1$ subsumes $C_2$ towards the ontology. If this is the case, then the corresponding variables $x_1$ and $x_2$ are set to be equal. Moreover, the cost of the proposition evoking $C_1(x_1)$ is set to zero. The resulting logical form is output to the optimized *Mini-TACITUS* system, which constructs its best interpretation using abductive axioms. The cost reduction based on semantic similarity is applied to those of the propositions in the interpretation, which have non-zero costs.

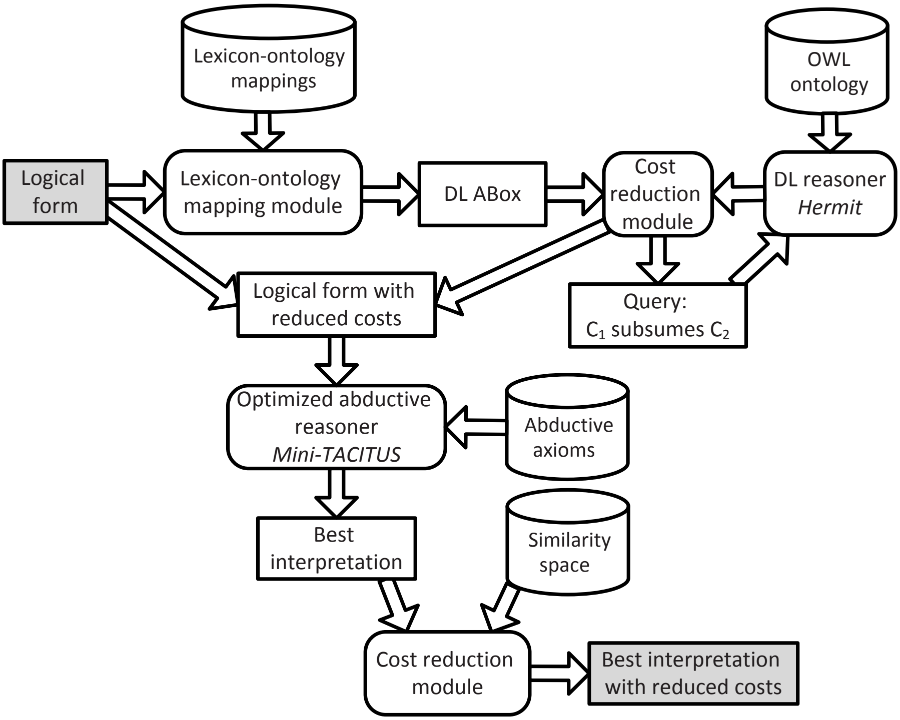


Fig. 7.1 Abductive reasoning pipeline.

# Chapter 8

# Evaluation

In this chapter, we present an evaluation of how well the developed integrative knowledge base (see chapters 5 and 6) and the proposed reasoning pipeline (Chap. 7) perform in different natural language understanding tasks.

As mentioned in Sec. 2.1, there are two strategies for evaluating NLU systems. The first strategy is application-based. An NLU system is requested to solve some of the tasks requiring text understanding, which can be faced by humans, e.g., answering questions about an input text, summarizing the text, etc.[1] Such an evaluation is definitely very useful, however, it poses several problems.

Interpreting a text fragment means solving many different small understanding tasks, which are highly interrelated. Recall the first example in this book.

> *"Romeo and Juliet" is one of Shakespeare's early tragedies. The play has been highly praised by critics for its language and dramatic effect.*

Understanding this short text fragment requires the inference of an implicit predicate in the possessive *Shakespeare's early tragedy*, resolving the anaphoric *it*, interpreting the relation between *early* and *tragedy*, relating *tragedy* and *play*, etc. Failing in one of these tasks can lead to overall failure even if other semantic phenomena were resolved correctly. Suppose that the anaphoric *it* in the second sentence was correctly linked to the preceding *play*, but *play* was not linked to *tragedy* and *"Romeo and Juliet"*. In this case, the system will be unable to infer that "Romeo and Juliet" has been praised by critics. Thus, an application-based evaluation forces an NLU system to be good enough at resolving all possible semantic phenomena.

[1] This evaluation strategy is realized in such series of test challenges as, for example, Text Analysis Conference (TAC, *http://www.nist.gov/tac/*).

Obviously, principles of designing texts used for an evaluation can crucially influence the results. For example, if a test set contains a lot of anaphoric expressions, but just a few cases of metonymy, then those systems, which are good at resolving anaphora will get better results, and the systems, which are only good at resolving metonymy will fail. Given another test set containing a lot of metonyms and a few anaphoric expressions, the situation will change. Organizers of the existing NLU competitions such as Question Answering or Recognizing Textual Entailment[2] have not yet developed a methodology for making sure that text corpora selected for an evaluation are semantically balanced, i.e. they reasonably represent a wide range of semantic phenomena.

The second evaluation strategy takes into account the internal representation of the text content created by the system. Here, an automatic annotation of a text performed by an NLU system is compared with a human annotation of the same text.[3] This approach allows us to focus on particular semantic phenomena. However, it also has its problems. Any semantic annotation necessarily depends on an underlying semantic theory and requires a particular representation format. Only those NLU systems, which do not conflict with this theory and can support the required annotation format can be evaluated using the corresponding annotated test corpus. Obviously, there is no guarantee that the underlying semantic theory gives an explanation of the described semantic phenomena, which is useful in practice. For example, the competitions on word sense disambiguation have been traditionally relying on WordNet as a reference source. However, many researchers have noticed that WordNet is much too fine-grained and even human annotators sometimes have difficulties in distinguishing between WordNet senses (see, for example, Agirre and Lacalle, 2003).

Although both evaluation strategies are not perfect, they both reveal interesting aspects of the evaluated systems and provide a possibility of comparing systems performing the same task. In this study, we follow both evaluation approaches. As an application-based evaluation, we use the recognizing textual entailment task. Representation-based evaluation was done by performing semantic role labeling and paraphrasing noun dependencies. The three tasks and the corresponding test datasets are described in Sec. 8.1.

On the early stage of the study, we performed a couple of experiments using the semantic parser *Boxer* and the deductive reasoning system *Nutcracker*. Although particularly promising results have not been obtained in these experiments, we document them in this

[2] *http://www.nist.gov/tac/tracks/index.html*

[3] This evaluation strategy is realized, for example, in the series of competitions called Semantic Evaluation (SemEval, *http://aclweb.org/aclwiki/index.php?title=SemEval_Portal*).

chapter (Sec. 8.2), because they reveal several interesting features of the corresponding approach.

Section 8.3 is devoted to the experiments performed using the abductive reasoning systems *Mini-TACITUS* adapted for being able to reason with the integrative knowledge base.

## 8.1 Natural Language Understanding Tasks

In this section, we describe the following natural language understanding tasks used for evaluation of the proposed approach:

- recognizing textual entailment,
- semantic role labeling,
- paraphrasing of noun dependencies.

### 8.1.1 *Recognizing Textual Entailment*

Recognizing textual entailment (RTE) is one of the recent tasks in NLP that captures major semantic inference needs in natural language understanding. Given two text fragments, namely, a *text* and a *hypothesis*, this task requires recognizing, whether the hypothesis can be inferred from the text. For example, given the text *"Romeo and Juliet" is a one of Shakespeare's early tragedies* and the two hypotheses *Shakespeare is the author of "Romeo and Juliet"* and *Shakespeare went through a tragedy*, the task is to predict that the first hypothesis is entailed by the text and the second one is not.

As noted by Bos and Markert (2005), recognizing textual entailment is similar to the famous Turing test indicating whether machines can think, because it presupposes understanding of language with access to linguistic and world knowledge as well as the ability to draw inferences. The RTE task can be seen as generic for several NLP applications. In information extraction, extracted information should be entailed by the corresponding text. In question answering, the answer is entailed by the supporting text fragment. In summarization, the text should entail its summary.

The notion of *entailment* referring to a relation between two text fragments was first proposed by Dagan and Glickman (2004). Although RTE has recently been attracting a lot of attention in the NLP community, there is still no unique formal definition of the task. Dagan *et al.* (2005) defined textual entailment as follows: "We say that a text T entails a hypothesis H if, typically, a human reading T would infer that H is most likely true". According to Dagan *et al.* (2010), this informal definition assumes common human understanding of language as well as common background knowledge.

In 2004, Dagan, Glickman, and Magnini have started a series of competitions under the PASCAL (Pattern Analysis, Statistical Modeling and Computational Learning[4]) Network of Excellence, known as the PASCAL Recognizing Textual Entailment Challenge (Dagan *et al.*, 2005). Since then the RTE Challenges take place every year proposing more and more complex modifications of the basic task. Starting from 2008, the challenge is organized within the Text Analysis Conference (TAC[5]) that accommodates a series of evaluation workshops providing large test collections, common evaluation procedures, and a forum to share the results. An overview of the first RTE Challenges is given, for example, in Dagan *et al.* (2010).

#### 8.1.1.1 *Experimental Data*

For RTE evaluation, we used the RTE-2 Challenge dataset[6] (Bar-Haim *et al.*, 2006). The RTE-2 dataset contains a development and a test set, both including 800 text-hypothesis pairs. Each dataset consists of four subsets, which correspond to typical success and failure settings in different applications: information extraction (IE), information retrieval (IR), question answering (QA), and summarization (SUM). In total, 200 pairs were collected for each application in each dataset.

The entailment pairs in the dataset were mostly generated by existing web-based systems; Bar-Haim *et al.* (2006) describe the data collection procedure in detail. The collected entailment pairs were judged manually, so that each pair was judged by at least two annotators. The average agreement on the test set (between each pair of annotators who shared at least 100 examples), was 89.2%. Additional filtering was done by two of the organizers, who removed pairs that seemed controversial, too difficult, or redundant. In total, 25.5% of the original pairs were removed from the test set. Minimal correction of texts extracted from the web was allowed, e.g., fixing spelling and punctuation. Table 8.1 contains some of the pairs in the RTE-2 development set.[7]

The main task in the RTE-2 challenge was entailment prediction for each pair in the test set. The evaluation criterion for this task was *accuracy* - the percentage of pairs correctly judged. The accuracy achieved by the 23 participating systems ranges from 53% to 75%.

[4] *http://www.pascal-network.org/*
[5] *http://www.nist.gov/tac/*
[6] *http://pascallin.ecs.soton.ac.uk/Challenges/RTE2/Datasets/*
[7] These examples are provided by (Bar-Haim *et al.*, 2006).

Table 8.1 Examples of text-hypothesis pairs from the RTE-2 development set.

| Text | Hypothesis | Task | Judgment |
|---|---|---|---|
| Google and NASA announced a working agreement, Wednesday, that could result in the Internet giant building a complex of up to 1 million square feet on NASA-owned property, adjacent to Moffett Field, near Mountain View. | Google may build a campus on NASA property. | SUM | YES |
| Drew Walker, NHS Taysides public health director, said: "It is important to stress that this is not a confirmed case of rabies." | A case of rabies was confirmed. | IR | NO |
| Meanwhile, in an exclusive interview with a TIME journalist, the first one-on-one session given to a Western print publication since his election as president of Iran earlier this year, Ahmadinejad attacked the "threat" to bring the issue of Irans nuclear activity to the UN Security Council by the US, France, Britain and Germany. | Ahmadinejad is a citizen of Iran. | IE | YES |
| About two weeks before the trial started, I was in Shapiros office in Century City. | Shapiro works in Century City. | QA | YES |

Two systems had 73% and 75% accuracy, two systems achieved 62% and 63%, while most of the systems achieved 55%–61% (cf. Bar-Haim *et al.*, 2006).

Garoufi (2007) has performed a detailed study of the RTE-2 dataset investigating factors responsible for entailment in a significant number of text-hypothesis pairs. Surprisingly, Garoufi's conclusion is that such shallow features as lexical overlap (number of words from hypothesis, which also occur in text) seem to be more useful for predicting entailment than any sophisticated linguistic analysis or knowledge-based inference. This fact may have two explanations: Either the RTE-2 dataset is not properly balanced for testing advanced textual entailment technology, or the state-of-the-art RTE systems indeed cannot suggest anything more effective than simple lexical overlap.

Nevertheless, we chose the RTE-2 dataset for our experiments. First, none of the other RTE datasets has been studied in so much detail, therefore there is no guarantee that any other dataset has better properties. Second, the RTE-2 test set was additionally annotated with FrameNet semantic roles which enables us to use it for evaluation of semantic role labeling.

### 8.1.2 *Semantic Role Labeling*

Semantic role labeling (SRL) is a task in natural language processing, which consists in the detection of semantic arguments associated with a predicate and the classification of the arguments into semantic roles. "Semantic roles represent the participants in an action or relationship captured by a semantic frame" (Gildea and Jurafsky, 2002). SRL provides a level of semantic interpretation useful for a number of NLP tasks. For example, identifying semantic roles would allow the system to determine that in the sentences *John broke the window with a stone*, *The window broke*, and *The stone broke the window* the subject of the first sentence is an agent, the subject of the second sentence is a patient, and the subject of the third sentence is an instrument. Thus, the first sentence in the example above entails the other two.

Classification of semantic arguments into semantic roles is performed using a predefined set of roles, cf. Sec. 3.1.1.2 for more details on different approaches to representing semantic arguments. In the experiments described in this book, we use the FrameNet inventory of frames and roles. With respect to FrameNet, assigning semantic roles is related to the disambiguation of predicates, because semantic roles are usually defined relative to a particular predicate sense. For example, in FrameNet the predicate *take* can have such roles as ACTIVITY and TIME_LENGTH in the sense of "taking time" (*the lecture took two hours*), and THEME and VEHICLE is the sense of "riding a vehicle" (*John took a bus*).

#### 8.1.2.1 *Experimental Data*

As a gold standard for semantic role labeling, we used the Frame-Annotated Corpus for Textual Entailment (FATE[8]), see Burchardt and Pennacchiotti (2008). This corpus provides manual FrameNet frame and role annotations for the RTE-2 challenge test set.

The main goal of the FATE annotation effort was to enable researchers to estimate the impact of FrameNet on the RTE task. It is important to note that FATE annotates only those constituents, which are *relevant* for computing entailment. The following example illustrates this principle.[9]

**T**: **Authorities** in Brazil <u>**say**</u> that more than 200 people are being **held** <u>**hostage**</u> in a **prison** in the **countrys remote**, Amazonian **jungle state** of Rondonia.

**H**: **Authorities** in Brazil **hold** 200 **people** as <u>**hostage**</u>.

[8] *http://www.coli.uni-saarland.de/projects/salsa*

[9] This example is provided by Burchardt and Pennacchiotti (2008).

Instead of annotating all words displayed in boldface above, which evoke a frame according to the FrameNet project, FATE only annotates the relevant frame evoking elements, which are underlined. According to FATE, these are only words, which evoke frames describing the overall situations in *H* and *T*. The developers of FATE admit that the notion of relevance in this context has a subjective character. After experimenting with experienced annotators reaching a high level of agreement (87%) they have formulated the following criterion of relevance: "a relevant [frame evoking element] evokes a situational frame and [...] instantiates in the text at least one role of the evoked frame" (Burchardt and Pennacchiotti, 2008).[10]

Similarly, for positive entailment pairs only those role fillers are annotated, which contribute to the inferential process that allows us to derive *H* from *T*. For negative pairs, it is often not possible to indicate, which portions of text contribute to a false entailment. Therefore, FATE annotates all possible role fillers in negative examples.

FATE has a special treatment of some linguistic constructions relevant for RTE, namely, support and copula verbs, existential construction, modal expressions, and metaphors (see Burchardt and Pennacchiotti, 2008, for more details). Moreover, the corpus introduces the ANAPHORA frame with the role ANTECEDENT, and the UNKNOWN frame used when the annotator did not find a relevant frame evoking element represented in the FrameNet hierarchy.

The corpus was annotated on top of the syntactic structure produced by the *Collins* parser (Collins, 1999), i.e. frame evoking elements and role fillers are syntactic constituents or sets of constituents as parsed by *Collins*.

The annotation was performed by one experienced annotator and checked for consistency using different strategies. First, a second experienced annotator was asked to annotate 5% of the corpus (40 *T*-*H* pairs). Agreement between the two annotators was calculated. *FEE-agreement* is the percentage of commonly annotated frame evoking elements. *Frame-agreement* is the percentage of commonly selected frames evoked by the same frame evoking element. *Role-agreement* is the percentage of commonly annotated roles (same name and same filler) belonging to commonly selected frames. The obtained agreements were: 82% FEE-agreement, 88% frame-agreement, 91% role-agreement. Second, the FATE developers have used sentences repeated across the RTE-2 dataset in order to calculate the self-agreement of the principal annotator. Based on 109 repetitions occurring in the dataset, the self-agreements were: 97% FEE-agreement, 98% a frame-agreement, and 88% role

[10] For more criteria see Burchardt and Pennacchiotti (2008).

Table 8.2 Evaluation of the *Shalmaneser* system towards FATE.

| **System** | **Frame match** | | **Role match** | | **Filler match** |
|---|---|---|---|---|---|
| | Precision | Recall | Precision | Recall | Accuracy |
| SHA | 0.48 | 0.55 | 0.54 | 0.37 | 0.77 |
| DET | 0.30 | 0.85 | 0.52 | 0.36 | 0.75 |

agreement. Burchardt and Pennacchiotti (2008) conclude that these results show a good level of consistency of the final annotation.

In total, 4 489 frames were annotated; 1 666 in the positive set and 2 823 in the negative set. The total number of annotated roles is 9 518; 3 516 in the positive set and 6 002 in the negative set. The annotation contains 373 instances of the UNKNOWN frame, which is 8% of the total frames. UNKNOWN roles are 1% of the total roles.

The FATE corpus was used as a gold standard for evaluating the *Shalmaneser* system (Erk and Pado, 2006), which is a state-of-the-art system for assigning FrameNet frames and roles. In Table 8.2, we replicate results for *Shalmaneser* alone (SHA) and *Shalmaneser* boosted with the WordNet Detour to FrameNet (DET). The WordNet-FrameNet Detour (Burchardt *et al.*, 2005) extended the frame labels assigned by *Shalmaneser* with the labels related via the FrameNet hierarchy or by the WordNet inheritance relation (cf. Burchardt *et al.*, 2009).

In frame matching, the number of frame labels assigned by the systems that occur in the gold standard annotation (precision) and the number of frame labels in the gold standard that can also be found in the system annotation (recall) were counted. Role matching was evaluated only on the frames that are correctly annotated by the system. The number of role labels found by the system, which also occur in the gold standard (precision) as well as the number of role labels in the gold standard annotation that can also be found in the system annotation (recall) were counted. Filler matching accuracy was computed on the set of correctly assigned roles only; it was defined as the percentage of fillers, which have identical syntactic head lemmas in automatic annotation and gold standard.

In the experiments described in Sec. 8.2.2 and 8.3.2 of this chapter, we compare the performance of the evaluated systems with the performance of *Shalmaneser*.[11] We ignore

[11]There exists one more probabilistic system labeling text with FrameNet frames and roles, which is called *SEMAFOR* (Das *et al.*, 2010). We do not compare our results with the results of *SEMAFOR*, because it has not been evaluated against the FATE corpus yet.

the ANAPHORA and UNKNOWN frames. We do not compare filler matching, because the FATE syntactic annotation produced by the *Collins* parser follows different standards as compared to the annotations produced by the semantic parsers used in the experiment, which makes aligning fillers non-trivial. Furthermore, since the evaluated systems make all possible frame assignments for a sentence, we provide only the recall measure for the frame match and leave the precision out.

### 8.1.3 *Paraphrasing of Noun Dependencies*

Noun dependencies considered in this book are

- noun compounds – "sequences of two or more nouns that act as a single noun" (Downing, 1977), e.g., *morning coffee*;
- genitive constructions – genitive noun phrases, which have an explicit genitive marker, 's, associated with them, e.g., *Shakespeare's tragedy*, and which are related via *of* preposition, e.g., *tragedy of Shakespeare*.

The problem of interpreting such dependencies consists in finding an appropriate relation explicitly expressing the dependency (cf. Butnariu *et al.*, 2010). For example, the noun compound *morning coffee* is most probably interpreted as a coffee drunk in the morning, while a possible interpretation of *morning newspaper* is "a paper read in the morning".

Experimental studies show that noun-noun constructions are very common and the compounding process is extremely productive in English (see, for example, Lapata and Lascarides, 2003a). Interpretation of noun dependencies is important for any NLU application, because it enables us to generate paraphrases (e.g., *Shakespeare's tragedy – tragedy written by Shakespeare*) which can be crucial for solving such tasks as question answering, information retrieval, and machine translation.

There are two main conceptions for interpretation of noun dependencies. The first conception assumes that a relation expressing the dependency can be represented in terms of a finite number of predefined predicates, such as *cause*, *have*, *make*, etc. (Levi, 1978), or prepositions *of*, *for*, *in*, *on*, etc. (Lauer, 1995), or semantic relations *possession*, *location*, *purpose*, etc. (Warren, 1978; Tratz and Hovy, 2010). This conception was realized in a SemEval evaluation challenge (Hendrickx *et al.*, 2010). In this evaluation setting, the task of interpretation of noun dependencies can be seen as a classification task with a predefined set of classes. Sentences are considered as contexts for the interpretation. Thus, the same noun-noun construction can be associated with different semantic predicates depending on these contexts. The main problem of this approach concerns the need for defining a finite set of semantic predicates, which is obviously the issue of constant disagreements between different researchers.

The second conception assumes that there is a possibly infinite number of predicates, which can express a particular noun dependency, such that these predicates correspond to specific natural language expressions (Downing, 1977; Nakov and Hearst, 2006; Butnariu *et al.*, 2010). For example, the verbs *drink* and *read* can express the implicit relations in the noun compounds *morning coffee* and *morning newspaper*. In an evaluation scenario based on this approach (Butnariu *et al.*, 2010), the system under evaluation is supposed to generate a set of paraphrases for each two nouns and score the paraphrases, so that the scores correlate well with human judgments. The problem with such an approach is that the interpretation of noun dependencies usually makes only sense in a context, so that the best interpretation is the one, which can be in the best way embedded into the context. Obviously, two nouns can imply multiple relations. For example, given the nouns *John* and *book*, the following relations can be suggested: *write*, *read*, *own*, *sell*. However, given the sentence *John's book was first published in 1922*, one will probably conclude that the "write" interpretation is the most probable one.

We intend to combine advantages of both approaches described above. As mentioned in the previous chapters of this book, a strong part of weighted abduction as applied to NLU is its ability to interpret text in context. We exploit this feature evaluating how well the procedure does with interpretation of noun dependencies in coherent text fragments using an open set of semantic relations expressed by natural language predicates.

#### 8.1.3.1 *Experimental Data*

For creating a test set, we manually investigated 1600 entailment pairs from the RTE-2 development and test sets. Only those noun compounds and possessives, which are responsible for the entailment inference were considered.

Consider the following entailment pair:

**T:** *Muslims make up some 3.2 million of* ***Germany's 82 million people****...*

**H:** ***82 million people live in Germany***.

In this pair, interpreting the possessive *Germany's 82 million people* as *82 million people live in Germany* is crucial for inferring entailment. In contrast, the possessive *city's airport* in the pair below does not contribute to the entailment inference:

**T:** *Two weeks ago, China became the first nation to operate a maglev railway commercially, when officials inaugurated a 30-kilometer-long line between downtown Shanghai and the* ***city's airport****.*

**H:** *Maglev is commercially used.*

Noun-noun constructions, which need not be expanded for inferring entailment were disregarded as well. For example, in the pair below there is no need to expand the possessive *John Lennon's widow* in order to infer the entailment, because it occurs both in *T* and in *H*.

**T:** ***John Lennon's widow****, Yoko Ono...*

**H:** *Yoko Ono is* ***John Lennon's widow****.*

The noun-noun constructions containing anaphora were also ignored, for example:

**T:** *Some 55 percent of the German public are opposed to the euro, less than 150 days before* ***its introduction****...*

**H:** ***The introduction of the euro*** *has been opposed.*

Since without an anaphora resolution procedure application of axioms to such phrases is useless, we did not considered them in the evaluation.

In total, 93 textual entailment pairs containing complex nominals relevant for inferring entailment have been found in the RTE-2 development (54) and test (39) collections. In the experiment described in Sec. 8.3.3, we measure for how many entailment pairs containing noun-noun constructions the procedure managed to infer the implicit predicate.

## 8.2 Experiments with *Boxer* and *Nutcracker*

In this section, we describe experiments performed with the semantic parser *Boxer* (see Sec. 4.1.2) and the recognizing textual entailment system *Nutcracker*, which uses *Boxer*'s output as an input for a deductive theorem prover and a model builder (see Sec. 4.2 and 8.2.1). Performing experiments with *Boxer* and *Nutcracker*, we aimed at investigating whether equipping these systems with additional lexical-semantic knowledge will improve their performance in NLU. First of all, we intended to experiment with the lexical-semantic knowledge extracted from WordNet and FrameNet. In order to enable *Nutcracker* to access this knowledge, the following two steps have been performed.[12]

[12]We especially thank Johan Bos for the fruitful collaboration concerning *Boxer* and *Nutcracker*.

*Disambiguation*

The *Boxer* system allows us to annotate lexemes with WordNet senses. However, the system does not have a disambiguation procedure. For each lexeme, it selects its most frequent sense according to WordNet-annotated corpora. This is, obviously, not the optimal solution, because all non-frequent word senses are permanently ignored. In order to improve on it, we equipped *Boxer* with a disambiguation module.

For performing disambiguation, we chose the *SenseLearner* system[13], which is a freely available word sense disambiguation (WSD) tool (Mihalcea and Faruque, 2004). *SenseLearner* is a statistical WSD system trained on the WordNet annotated corpus SemCor[14]. The system has participated in the Senseval-3 WSD evaluation[15] and obtained an average accuracy of 64.6% given the most frequent sense baseline of 60.9%. As most of the WSD systems, *SenseLearner* has a relatively good performance on nouns and significantly worse results for verbs (cf. Mihalcea and Faruque, 2004). The reason for this may be that verb senses in WordNet are particularly fine-grained which complicates an automatic discrimination of senses (see, for example, Agirre and Lacalle, 2003). Therefore, we applied *SenseLearner* disambiguation procedure to nouns only.

*Assignment of FrameNet frames and roles*

As mentioned in Sec. 4.1.2, *Boxer* is able to map predicates and their arguments into VerbNet classes and semantic roles. For example, *Boxer* will produce the following FOL logical form for the sentence *John gave Mary a book*, where *Agent*, *Recipient*, and *Theme* are VerbNet roles, and 13.1 is the VerbNet class assigned to *give*.

$$\exists x_1, x_2, x_3, x_4 (John(x_1) \wedge Mary(x_2) \wedge book(x_3) \wedge give(x_4, 13.1) \wedge$$
$$Agent(x_4, x_1) \wedge Recipient(x_4, x_2) \wedge Theme(x_4, x_3))$$

We used this feature of *Boxer* for annotating the predicate-argument constructions with FrameNet frames and frame roles, which enables us to apply frame relations. For this sake, we used the *SemLink*[16] mapping between VerbNet and FrameNet, which maps VerbNet classes and roles into FrameNet frames and roles. For example, the VerbNet class `give-13.1` is mapped to the FrameNet frame GIVING, so that the role *Agent* is mapped

[13] *http://lit.csci.unt.edu/∼senselearner/*
[14] *http://www.cse.unt.edu/∼rada/downloads.html#semcor*
[15] *http://www.senseval.org/senseval3/*
[16] *http://verbs.colorado.edu/semlink/*

to DONOR, *Theme* is mapped to THEME, and *Recipient* is mapped to RECIPIENT. Such mapping allows us to convert the logical form above into the following.

$$\exists x_1, x_2, x_3, x_4 (John(x_1) \land Mary(x_2) \land book(x_3) \land give(x_4, 13.1) \land$$
$$Giving(x_4) \land Donor(x_4, x_1) \land Recipient(x_4, x_2) \land Theme(x_4, x_3))$$

FrameNet relations were converted into FOL axioms following the axiomatization presented in Sec. 5.2.2. These axioms constituted the knowledge base for the *Nutcracker* system. For example, the relation between the frames GIVING and GETTING was converted into the following axioms.

$$\forall x, y(\text{GIVING}(x) \rightarrow \text{GETTING}(y) \land causes(x, y))$$
$$\forall x, y, z((\text{GIVING}(x) \land \text{GETTING}(y) \land causes(x, y)) \rightarrow (\text{THEME}(x, z) \leftrightarrow \text{THEME}(y, z)))$$
$$\forall x, y, z((\text{GIVING}(x) \land \text{GETTING}(y) \land causes(x, y)) \rightarrow (\text{DONOR}(x, z) \leftrightarrow \text{SOURCE}(y, z)))$$
$$\forall x, y, z((\text{GIVING}(x) \land \text{GETTING}(y) \land causes(x, y)) \rightarrow$$
$$(\text{RECIPIENT}(x, z) \leftrightarrow \text{RECIPIENT}(y, z)))$$

The overall NLU pipeline based on *Boxer* and *Nutcracker* is shown in Fig. 8.1. A text fragment is first input to the *Boxer* semantic parser that produces its logical form (DRS) in a FOL format (see Sec. 4.1.2). The DRS is processed by the disambiguation module, which is based on the *SenseLearner* tool. Then, the DRS structure is assigned FrameNet frames and roles using the *SemLink* mapping. The resulting DRS is used to evaluate semantic role labeling. Furthermore, the DRSs are processed by the recognizing textual entailment system *Nutcracker* using the knowledge base consisting of WordNet synonymy and hyperonymy relations and FrameNet frame relations. The system employs the FOL theorem prover *BLIKSEM* and the FOL model builders *MACE* and *Paradox* (cf. Sec. 4.2).

### 8.2.1 *Recognizing Textual Entailment*

Given a text $T$ and a hypothesis $H$, *Nutcracker* is able to perform different types of inferences. Assume that the function `DRS` translates $T$ and $H$ into logical forms (DRSs) represented in FOL (cf. Sec. 4.1.2). If the theorem prover manages to prove

$$\mathtt{DRS}(T) \rightarrow \mathtt{DRS}(H),$$

then *Nutcracker* concludes that the text entails the hypothesis. Furthermore, if the theorem prover finds a prove for

$$\neg\,(\mathtt{DRS}(T) \land \mathtt{DRS}(H)),$$

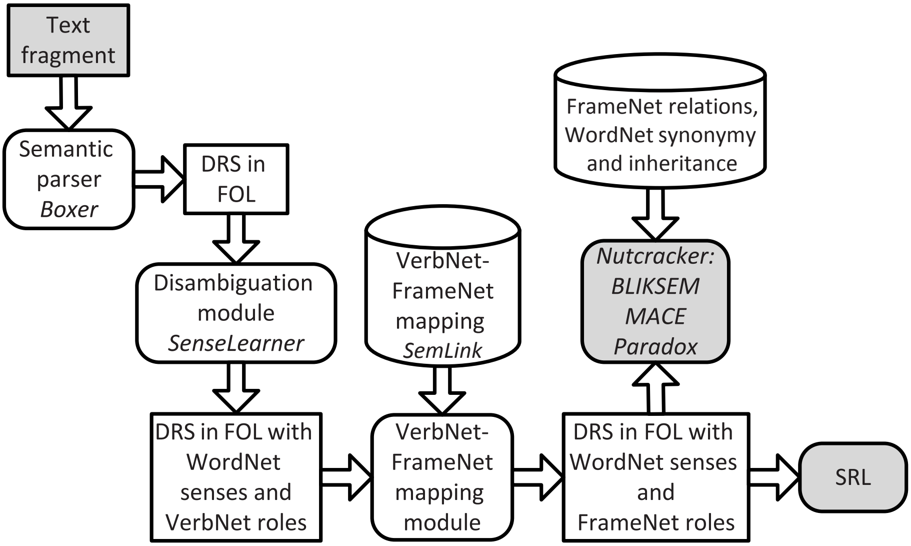


Fig. 8.1 NLU pipeline based on *Boxer* and *Nutcracker*.

then *Nutcracker* concludes that $T$ and $H$ are inconsistent ($T$ and $H$ logically contradict each other) and $T$ definitely does not entail $H$.

For example, the theorem prover will find that $T$ entails $H$ for the following $T$-$H$ pair.[17]

**T**: *His family has steadfastly denied the charges.*

**H**: *The charges were denied by his family.*

*Nutcracker* allows us introducing a knowledge base represented by FOL axioms. If a knowledge base *KB* is specified, the system will use it trying to prove the following formulas:

$$(KB \land \texttt{DRS}(T)) \rightarrow \texttt{DRS}(H)$$
$$\neg\, (KB \land \texttt{DRS}(T) \land \texttt{DRS}(H))$$

*Nutcracker* is supplied with a default knowledge base consisting of the axioms derived from the WordNet hypernymy and synonymy relations as follows: If a word sense $w_1$ occurs in a synset $S_1$, a word sense $w_2$ occurs in $S_2$, and the relation is defined on $S_1$ and

[17]This and the following examples in this section are provided by Bos and Markert (2005).

$S_2$, then the axiom $\forall x(w_1(x) \rightarrow w_2(x))$ is introduced. *Boxer* allows mappings of lexemes into WordNet senses. Thus, given logical forms produced by *Boxer*, direct application of WordNet axioms is possible. For example, the following *T-H* pair can be resolved using WordNet, since the verbs *soar* and *rise* belong to the same synset.

**T**: *Crude oil prices **soared** to record levels.*

**H**: *Crude oil prices **rise**.*

*Nutcracker* combines a theorem prover with a model builder. The theorem prover attempts to prove the input, while the model builder tries to find a model for the negation of the input. If the model builder finds a model for the formula

$$\neg\,(\mathtt{DRS}(T) \rightarrow \mathtt{DRS}(H)),$$

then there cannot be an entailment between $T$ and $H$. If the model builder is able to generate a model for

$$\mathtt{DRS}(T) \wedge \mathtt{DRS}(H),$$

then $T$ and $H$ are consistent and entailment is possible. Furthermore, *Nutcracker* uses the difference in domain size of the models (number of individuals in the models) generated for $\mathtt{DRS}(T)$ and $\mathtt{DRS}(T) \wedge \mathtt{DRS}(H)$ as a heuristic for predicting entailment. The intuition behind it is that if $H$ is entailed by $T$, then the model for $\mathtt{DRS}(T) \wedge \mathtt{DRS}(H)$ is not informative compared to the one for $\mathtt{DRS}(T)$, and hence does not introduce new entities. In addition to domain size, *Nutcracker* relies on comparing model sizes of $KB \wedge \mathtt{DRS}(T)$ and that of $KB \wedge \mathtt{DRS}(T) \wedge \mathtt{DRS}(H)$. Model size is defined by counting the number of all instances of two-place relations (and three-place relations, if there are any) in the model, and multiplying this with the domain size. The underlying idea is that if the difference of the model sized is small, it is likely to be an entailment.

In addition to the described deep semantic features, *Nutcracker* uses such shallow features as lexical overlap (the number of words from $H$ occurring in $T$), length of text and hypothesis, and overlap of the WordNet synsets evoked by text and hypothesis (see Bos and Markert, 2005, for more details).

*Nutcracker* was evaluated on the RTE-2 Challenge dataset (Bos and Markert, 2005). The run using only the shallow features yielded an accuracy of 61.6%, and the run using both shallow and deep features performed with an accuracy score of 60.6%. The system managed to find 29 proofs, of which 19 proofs were found without a knowledge base and 10 proofs were found with a small manually created knowledge base.

We ran *Nutcracker* on the same dataset. The system was equipped with the disambiguation tool *SenseLearner* and axioms derived from FrameNet, as described in the previous section. The results did not appear to be encouraging. Disambiguation did not result in any improvements. Only 1 additional proof was found with the help of FrameNet axioms. We hypothesized that one of the reasons for it might be the insufficient quality of the semantic role labeling procedure assigning frames and frame roles (which was indeed the case, cf. the next section). In order to understand whether more proofs will be found using a better semantic role annotation, we extracted the FrameNet frame and role labeling from the manually annotated FATE corpus (see Sec. 8.1.2.1) and automatically added it to the *Boxer*'s output. Running *Nutcracker* on *Boxer*'s output enriched with the FATE annotation gave only 3 more proofs.

The reason for the overall failure of the experiment seems to be the inability of the deductive theorem prover to reason given an incomplete knowledge base. Obviously, axioms from WordNet and FrameNet cannot cover all information needed for English text interpretation. In the cases when it is impossible to provide it with *all* the knowledge, which is relevant for interpretation of a particular piece of text, the deductive theorem prover will fail to find a proof.

Consider the example below replicating the FATE annotation. In the text of this example, the noun phrase *joint venture* evokes the frame ALLIANCE, which has the roles MEMBER_1 and MEMBER_2 filled by *Botswana* and *De Beers*. In the hypothesis, the noun *partner* evokes the frame COLLABORATION with the roles PARTNER_1, PARTNER_2, and UNDERTAKING (this role marks constituents that express the project in which the partners are collaborating). A frame relation can be used to relate the ALLIANCE frame to COLLABORATION mapping MEMBERs to PARTNERs. But what to do with *business* filling the UNDERTAKING role? Our common sense knowledge suggests that venture is necessarily related to business. However, if the corresponding axiom is missing from the knowledge base then the theorem prover will fail to find a proof.

**T**: ...[*Botswana's*]$_{\text{MEMBER\_1}}$ [*50-50*]$_{\text{DESCRIPTOR}}$ [*joint venture*]$_{\text{ALLIANCE}}$ [*with De Beers*]$_{\text{MEMBER\_2}}$...

**H**: [*Botswana*]$_{\text{PARTNER\_1}}$ is a [*business*]$_{\text{UNDERTAKING}}$ [*partner*]$_{\text{COLLABORATION}}$ [*of De Beers*]$_{\text{PARTNER\_2}}$.

In practice, knowledge gaps occur frequently. What one would like to have instead of a deterministic "yes" or "no" proof is a way of measuring in how far the input formula was proven and which of its parts could not be proven. This option has not been implemented in the existing deductive reasoners yet. In order to overcome the described problem, RTE systems based on deduction employ two solutions. Some systems introduce different heuristics and consider logical inference to be just one of the features used to classify the *T*-*H* pair as entailment or not entailment (Bos and Markert, 2005). Other systems decompose logical forms into atomic propositions and measure whether each proposition from *H* is entailed by a proposition from *T* (Akhmatova and Mollá, 2005; Fowler *et al.*, 2005; Tatu and Moldovan, 2005). Based on this solutions, a measure of "entailmentness" can be calculated.

#### 8.2.1.1 *Case Study*

Since the quantitative evaluation of the knowledge base did not give particularly interesting results, we performed a qualitative analysis by investigating a limited number of the RTE-2 entailment pairs in detail. Section 6.1 describes a case study on cleaning up "medical" frames according to the ontological principles proposed in this book. It is important to note that the frames were cleaned up manually, which resulted in high quality axioms. For investigating the impact of the restructured network of "medical" frames (see Fig. 6.2) on RTE, we selected the *T*-*H* pairs from the FATE corpus such that both *T* and *H* are annotated with one of the "medical" frames. In total, 39 entailment pairs were selected. Textual entailment was computed a) without frame annotation (NFA), b) with frame annotation but without frame relation axioms (FA), c) with frame relations axiomatized as shown in Fig. 6.2 (bottom) (FA&A).

Only one correct proof was found without employing frame annotation. Then, as it was done in the previous experiment, frame and frame role annotation was imported from FATE. This time it was done manually in order to avoid errors arising from automatic mapping. Adding the FATE annotation manually allowed us to prove 3 more entailments. 3 more proofs were found with the help of the frame relations. Manually going through the *T*-*H* pairs used in the experiment we discovered that finding a proof failed in 12 cases because of a) incompleteness of the FATE annotation (8 pairs), b) *Boxer*'s processing errors (5 pairs), c) lack of general non-definitional knowledge (7 pairs). The last observation corresponds to the findings reported by Clark *et al.* (2007). The authors have considered 100 positive entailment pairs from the RTE-3 set and concluded that the majority of the *T*-*H* pairs require complex knowledge of the non-definitional type for their resolution. For example, arriving at a hospital usually means being hospitalized, suffering from a disease

Table 8.3 Results of recognizing textual entailment by the *Nutcracker* system for the 39 RTE-2 pairs annotated with “medical” frames in FATE.

| | **NFA** | **FA** | **FA&A** |
|---|---|---|---|
| Correct proofs | 1 | 4 | 7 |
| Wrong proofs | 1 | 1 | 1 |
| Overall accuracy | 0.56 | 0.5 | 0.61 |

Table 8.4 Evaluation of the *Boxer* system performing SRL towards FATE.

| **System** | **Frame match** | **Role match** | |
|---|---|---|---|
| | Recall | Precision | Recall |
| *Boxer* | 0.12 | 0.82 | 0.32 |

typically means having the disease, etc. Obviously, FrameNet alone is not sufficient to provide this knowledge. The final results are shown in Table 8.3.

Although the presented case study cannot be considered as an evaluation of the FrameNet axioms, we can conclude that restructuring and enriching frame relations according to the methodology proposed in Sec. 6.1 gives advantages in NLU. Unfortunately, cleaning up frames implies a lot of manual work. Further work in this direction includes developing automatic procedures for detecting and repairing conceptual inconsistencies.

### 8.2.2 *Semantic Role Labeling*

Since the application of the *SemLink* mapping from VerbNet to FrameNet results in labeling predicate-argument constructions with frames and roles, we additionally evaluated the quality of the obtained semantic role labeling using FATE as a gold standard as described in Sec. 8.1.2.1. The obtained results are again not very promising (see Table 8.4; the recall and precision measures are described in Sec. 8.1.2.1). Only a small number of frames from the gold standard (12%) were found by the system.

The SRL procedure in *Boxer* consists of 2 steps relying on different resources: 1) mapping of predicate-argument constructions to VerbNet classes and roles by *Boxer* and 2) mapping of VerbNet classes and roles into FrameNet frames and roles using *SemLink*. Each of these steps introduces errors. First, *Boxer* has no disambiguation procedure. Verbs are mapped to the most frequent appropriate VerbNet class. Second, the *SemLink* mapping is incomplete. It concerns only those lexical items, which occur both in VerbNet and in FrameNet. For example, the verb *sling* (VerbNet class `throw-17.1-1`) does not occur in FrameNet. However, it could possibly be mapped to the CAUSE_MOTION frame. Similarly, the words occurring in FrameNet alone are ignored. Moreover, *SemLink* misses some important role links. For example, *SemLink* links the VerbNet class `die-48.2` to the FrameNet frame DEATH. However, it does not link the role *Source* to the CAUSE role. Therefore, the sentence *John died from cancer*, where *cancer* fills the *Source* role, cannot be properly annotated with FrameNet roles. Unfortunately, all these issues make the SRL procedure based on the current versions of *Boxer* and *SemLink* less than promising.

## 8.3 Experiments with *Mini-TACITUS*

In this section, we describe a series of experiments performed using the abductive reasoner *Mini-TACITUS*. In these experiments, the *ESG* semantic parser was employed (see Sec. 4.1.1).[18] *Mini-TACITUS* was extended as described in Chap. 7. The knowledge base used for reasoning is described in Chap. 5.

The overall NLU pipeline based on *ESG* and *Mini-TACITUS* is shown in Fig. 8.2. An input text is processed by the *ESG* semantic parser, which outputs logical forms. These logical forms together with the knowledge base are used by the reasoning pipeline based on *Mini-TACITUS*, which is described in Sec. 7.5. The knowledge base consists of a) the axioms extracted from WordNet, FrameNet, PropBank, b) the similarity space, and c) the car ontology. The resulting best interpretation is used by three applications: recognizing textual entailment, semantic role labeling, and paraphrasing of noun-noun constructions.

As mentioned in Sec. 4.3, weighted abduction relies on the following heuristics:

- the cost function $f$,
- costs of the input propositions,
- non-merge constraints,
- axiom weights.

As mentioned in Sec. 5.1.2, we use the cost function $f(w,c) = w \cdot c$, such that $w$ is the assumption weight and $c$ is the cost of the input proposition. Let $c$ be the cost of the proposition $b$ occurring in the input logical form and $w$ be the weight of $a$ in the axiom $a^w \rightarrow b$. If $b$ costs 20 and the weight $w$ is equal to 1.2 then assuming $a$ costs 24.

[18] We thank Michael McCord for making the *ESG* parser available.

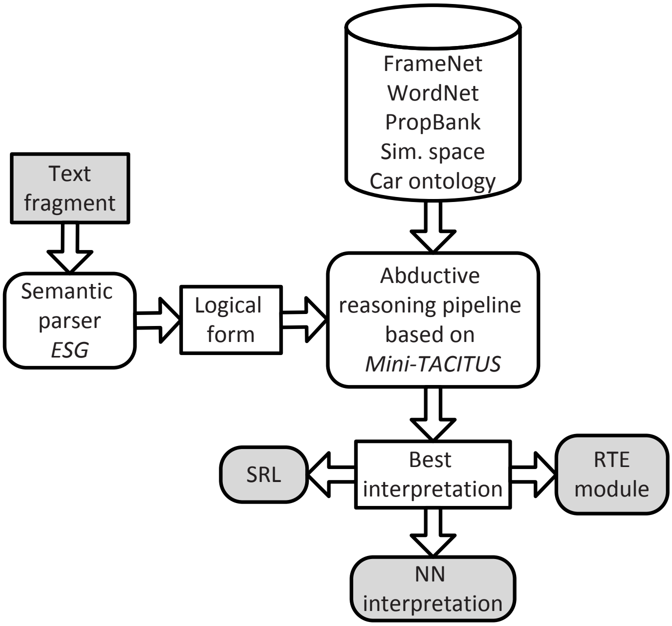


Fig. 8.2 NLU pipeline based on *ESG* and *Mini-TACITUS*.

Given the cost function defined above, the actual value of the costs of the input propositions does not matter, because the axiom weights, which affect the costs of the resulting interpretations, are given as percentages of the input proposition costs (see the cost function $f$ defined above). The only heuristic we use here concerns setting all costs of the input propositions to be equal.

Non-merge constraints used in the presented experiments are described in Sec. 7.2. Axiom weights for lexical-semantic axioms are based on frequencies as described in Sec. 5.2.1, 5.2.2, and 5.2.3.

### 8.3.1 *Recognizing Textual Entailment*

Our approach to recognizing textual entailment is to interpret both the text and the hypothesis using weighted abduction, and then see whether adding information derived from the text to the knowledge base will reduce the cost of the best abductive proof of the hypothesis as compared to using the original knowledge base only.[19] If the cost reduction exceeds a threshold determined from a training set, then we predict entailment.

A simple example would be the text *John gave a book to Mary* and the hypothesis *Mary got a book*. Our pipeline constructs the following logical forms for these two sentences; the numbers (20) after every proposition correspond to the default costs of the input propositions.

**T**: $John(e_1,x_1):20 \wedge give(e_2,x_1,u_1,x_3):20 \wedge book(e_3,x_3):20 \wedge$
$to(e_4,e_2,x_2):20 \wedge Mary(e_5,x_2):20$

**H**: $Mary(e_1,x_1):20 \wedge get(e_1,x_1,x_2,u_1):20 \wedge book(e_3,x_2):20$

These logical forms constitute the *Mini-TACITUS* input. *Mini-TACITUS* applies the axioms from the knowledge base to the input logical forms in order to reduce the overall cost of the interpretations. Suppose that we have three FrameNet axioms in our knowledge base[20]:

(1) $\textsc{Giving}(e_0,x_1,x_2,x_3)^{0.9} \rightarrow give(e_0,x_1,u_1,x_3) \wedge to(e_1,e_0,x_2)$
(2) $\textsc{Getting}(e_0,x_1,x_2,u_1)^{0.9} \rightarrow get(e_0,x_1,x_2,u_2)$
(3) $\textsc{Giving}(e_0,x_1,x_2,x_3)^{1.2} \rightarrow \textsc{Getting}(e_1,x_2,x_3,x_1)$

Axiom (1) maps *give to* to the GIVING frame, so that the second argument of *give* refers to the DONOR role and its fourth argument refers to the THEME role. The filler of the RECIPIENT role is expressed by the prepositional phrase, therefore the third argument $x_4$ of *give* is uninstantiated ("dummy"). Axiom (2) maps *get* to GETTING, so that the filler of the SOURCE role remains implicit. For the sake of simplicity, the role names are omitted in axioms (1) and (2). The second, third, and fourth argument positions of the GIVING frame correspond to the DONOR, RECIPIENT, and THEME roles. The second, third, and fourth argument positions of the GETTING frame correspond to the RECIPIENT, THEME, and SOURCE roles. Axiom (3) relates the frames GIVING and GETTING, so that the corresponding frame roles are mapped as follows: DONOR – SOURCE ($x_1$), RECIPIENT – RECIPIENT ($x_2$), THEME – THEME ($x_3$). The first two axioms have the weights of 0.9 and the third one has the weight of 1.2.

As a result of the application of the axioms to the logical forms of *T* and *H* the following best interpretations will be constructed:

[19] This approach was first presented in Ovchinnikova *et al.* (2011).
[20] The numbers after each proposition correspond to the weight of this proposition being assumed.

**I(T)**: $John(e_1,x_1):20 \wedge give(e_2,x_1,u_1,x_3):0 \wedge book(e_3,x_3):20 \wedge$
$to(e_4,e_2,x_2):20 \wedge Mary(e_5,x_2):20 \wedge \textsc{Giving}(e_2,x_1,x_2,x_3):18$

**I(H)**: $Mary(e_1,x_1):20 \wedge get(e_2,x_1,x_2,u_1):0 \wedge book(e_3,x_2):20 \wedge$
$\textsc{Getting}(e_2,x_1,x_2,u_2):18$

Note that $give(e_2,x_1,u_1,x_3)$ and $to(e_4,e_2,x_2)$ were backchained on to $\textsc{Giving}(e_2,x_1,x_2,x_3)$ in $T$ and $get(e_2,x_1,x_2,u_1)$ was backchained on to $\textsc{Getting}(e_2,x_1,x_2,u_2)$ in $H$. The total cost of the best interpretation for $H$ is equal to 58. Now the best interpretation of $T$ will be added to $H$ with zero costs (as if $T$ was totally proven) and we will try to prove $H$ once again. First of all, merging of the propositions with the same names will result in reducing costs of the propositions *Mary* and *book* to 0, because they occur in $T$:

**I(T+H)**: $John(e_1,x_1):0 \wedge give(e_2,x_1,u_1,x_3):0 \wedge book(e_3,x_3):0 \wedge$
$to(e_4,e_2,x_2):0 \wedge Mary(e_5,x_2):0 \wedge \textsc{Giving}(e_2,x_1,x_2,x_3):0 \wedge$
$get(e_6,x_2,x_3,u_2):0 \wedge \textsc{Getting}(e_6,x_2,x_3,u_3):18$

The only proposition left to be proved is GETTING. Using the GETTING-GIVING relation expressed by axiom (3) above, this proposition can be backchained on to GIVING, which will merge with GIVING coming from the $T$ sentence.

**I(T+H)**: $John(e_1,x_1):0 \wedge give(e_2,x_1,u_1,x_3):0 \wedge book(e_3,x_3):0 \wedge$
**best** $to(e_4,e_2,x_2):0 \wedge Mary(e_5,x_2):0 \wedge \textsc{Giving}(e_2,x_1,x_2,x_3):0 \wedge$
$get(e_6,x_2,x_3,x_1):0 \wedge \textsc{Getting}(e_6,x_2,x_3,x_1):0$

$H$ appears to be proven completely with respect to $T$; the total cost of its best interpretation given $T$ is equal to 0. Thus, using knowledge from $T$ helped to reduce the cost of the best interpretation of $H$ from 58 to 0.

The approach presented does not have any special account for logical connectors such as *if*, *not*, *or*, etc. Given a text *If A then B* and a hypothesis *A and B* our procedure will most likely predict entailment. Even more problematic is the negation issue. Thus, *not A* entails *A* in this approach. At the moment our RTE procedure mainly accounts for the informational content of texts, being able to detect the "aboutness" overlap of $T$ and $H$. In our framework, a fuller treatment of the logical structure of natural language would presuppose a more complicated strategy of merging redundancies.

We evaluated our procedure on the RTE-2 dataset described in Sec. 8.1.1.1. The RTE-2 development set was used to train the threshold for discriminating between the "entailment" and "no entailment" cases.[21] As a baseline we processed the datasets with an empty knowledge base. The depth parameter was set to 3 (see Sec. 7.1).

Then, we have done different runs, evaluating knowledge extracted from different resources separately.[22] Table 8.5 contains results of our experiments, where "Max. proc. time" stands for the value of the time parameter – maximal processing time per sentence in minutes (see Sec. 7.1); "Number of axioms" stands for the average number of axioms per sentence. The experiments were performed with an empty knowledge base (No KB), axioms extracted from WordNet 3.0 (WN 3.0), axioms extracted from FrameNet (FN), axioms extracted from definitions in extended WordNet 2.0 (Ext. WN 2.0), axioms for interpretation of noun dependencies extracted from the Proposition Store (NN), similarity space ($SIM^1$ and $SIM^2$), and the combination of WN 3.0, FN, NN, and $SIM^1$ resources.

The obtained baseline of 57.3% is close to the lexical overlap baselines reported by the participants of RTE-2 (Bar-Haim *et al.*, 2006). The NN knowledge resource did not give any significant improvement, because the interpretation of noun-noun combinations appeared to be not sufficient for predicting entailment in most of the cases (cf. Sec. 8.3.3). Although FrameNet has provided less axioms than WordNet in total (ca. 50 000 vs. 600 000), its application resulted in better accuracy than application of WordNet. The reason for this might be the confusing fine-grainedness of WordNet which makes word sense disambiguation difficult. Moreover, the average number of WordNet axioms per sentence is smaller than the number of FrameNet axioms (cf. Table 8.5). This happens because the relational network of FrameNet is much more dense than that of WordNet.

In Sec. 7.4, two different options for employing semantic similarity in the abductive reasoning procedure were proposed. The first option implies measuring how much each proposition $p$ is implied by the whole context ($SIM^1$ run). Using the second option, we try to find at least one strong connection between $p$ and some other proposition $p'$ in the logical form. The first run ($SIM^1$) gave a better performance of 62.2% (cf. Table 8.5). Semantic similarity performed better than axioms, which corresponds to the observations made by Garoufi (2007) claiming that most of the inferences in the RTE-2 dataset could be made on the basis of lexical overlap and similarity. A similarity space calculated using a more up-to-date corpus would definitely give better results. Unfortunately, the similarity space

[21]Interpretation costs were normalized to the number of propositions in the input. This was done in order to normalize over the prediction of longer and shorter hypotheses. If a hypothesis $h_1$ contains more propositions than $h_2$, then it can potentially contain more propositions not linked to propositions in the text.

[22]The computation was done on a High Performance Cluster (320 2.4 GHz nodes, CentOS 5.0) of Center for Industrial Mathematics (Bremen, Germany). We thank Theodore Alexandrov for making the cluster available.

used in the described experiments was based on a quite outdated corpus (cf. Sec. 5.4), so that many of the proper names relevant for RTE-2 were missing.

The lower performance of the system using the KB consisting of axioms extracted from extended WordNet (Ext. WN 2.0) can be explained. The axioms extracted from the synset definitions introduce a lot of new lexemes into the logical form, since these axioms define words with the help of other words rather than abstract concepts. These new lexemes trigger more axioms. Finally, too many new lexemes are added to the final best interpretation, which can often be noisy. The WN 3.0 and FN axioms set do not cause this problem, because these axioms operate on frames and synsets rather than on lexemes.

The best result was obtained when using the WN 3.0, FN, NN, and SIM[1] resources in combination. This run gives 64.1% accuracy. Recall that two systems participating the RTE-2 Challenge had 73% and 75% accuracy, two systems achieved 62% and 63%, while most of the systems achieved 55%-61% (Bar-Haim *et al.*, 2006). Thus, the obtained result allows us to conclude that the proposed procedure showed performance compatible with those of the state-of-the art systems specially designed for RTE.

For our best run (WN 3.0 + FN + NN + SIM[1]), we present the accuracy data for each application separately (Table 8.5 bottom). The distribution of the performance of *Mini-TACITUS* on the four datasets corresponds to the average performance of systems participating in RTE-2 as reported by Garoufi (2007). The most challenging task in RTE-2 appeared to be IE. QA and IR follow, and finally, SUM was titled the "easiest" task, with a performance significantly higher than that of any other task.

Experimenting with the time parameter $t$ restricting processing time (see Sec. 7.1) we found that the performance of *Mini-TACITUS* increases with increasing time of processing. This is not surprising. The smaller $t$ is, the fewer chances *Mini-TACITUS* has to apply all relevant axioms. Tracing the reasoning process, we found that given a long sentence and a short processing time *Mini-TACITUS* had time to construct only a few interpretations, and the "real" best interpretation was not always among them. For example, if the processing time is restricted to 30 minutes per sentence and the knowledge base contains some hundreds of axioms, then *Mini-TACITUS* has not enough time to apply all axioms and construct all possible interpretations in order to select the best one, while processing a single sentence for 30 minutes is definitely unfeasible in a realistic NLU setting. This suggests that optimizing the system computationally could lead to producing significantly better results.

Table 8.5 Evaluation of the *Mini-TACITUS* system performing RTE using RTE-2 test dataset.

| **KB** | **Accuracy** | **Max. proc. time** | **Number of axioms** | |
|---|---|---|---|---|
| | | | **T** | **H** |
| No KB | 57.3% | 1 | 0 | 0 |
| WN 3.0 | 59.6% | 30 | 294 | 111 |
| FN | 60.1% | 30 | 1233 | 510 |
| Ext. WN 2.0 | 58.1% | 30 | 215 | 85 |
| NN | 58.0% | 10 | 75 | 29 |
| $SIM^1$ | 62.2% | 10 | 0 | 0 |
| $SIM^2$ | 61.1% | 10 | 0 | 0 |
| WN 3.0 + FN + NN + $SIM^1$ | 64.1% | 40 | 1602 | 649 |

| **Task** | **Accuracy** |
|---|---|
| Summarization (SUM) | 78% |
| Information Retrieval (IR) | 66% |
| Question Answering (QA) | 62% |
| Information Extraction (IE) | 50% |

### 8.3.2 *Semantic Role Labeling*

Since our knowledge base contains axioms derived from FrameNet, which allow us to map predicate-argument constructions into frames (cf. Sec. 5.2.2), semantic role labeling was obtained as a by-product of constructing the best interpretation of the sentences from the RTE-2 test set. No special adjustment of the procedure to the SRL task was needed.

The only change with respect to the experiments on RTE concerns axiom weights. As stated in Sec. 5.2.2, all FrameNet axioms have weights ranging from 1 to 2, so that an axiom will be applied only if it can reveal a redundancy in the logical form, which will result in merging propositions and decrease of the interpretation cost. However, in order to achieve a higher recall in SRL, we would like to assign frames and roles to all possible parts of the logical form. Therefore, we changed weights of FrameNet axioms to make them rage from 0 to 1, so that any frame assignment would lower the cost of the frame evoking proposition.

Table 8.6 shows that given FrameNet axioms, the performance of *Mini-TACITUS* on semantic role labeling of the RTE-2 test set is compatible with the state-of-the-art system *Shalmaneser* specially designed to solve this task, compare with the results of the *Shal-*

Table 8.6 Evaluation of the *Mini-TACITUS* system performing SRL towards FATE.

| System | Frame match | Role match | |
|---|---|---|---|
| | Recall | Precision | Recall |
| *Mini-TACITUS* | 0.65 | 0.55 | 0.30 |

*maneser* system in Table 8.2.[23] As compared to the approach based on *Boxer* and *Semlink* (see Table 8.4), *Mini-TACITUS* assigns significantly more relevant frames (65% vs. 12%). However, given the correctly assigned frames, *Mini-TACITUS* is less accurate in assigning the semantic roles.

Unfortunately, FrameNet does not really provide any semantic typing for the frame roles. This type of information would be extremely useful for solving the SRL task. For example, consider the phrases *John took a bus* and *the meeting took 2 hours*. The lexeme *take* can be mapped both to the RIDE_VEHICLE and TAKING_TIME frame. Our system can use only the external context for disambiguation of the verb *take*. For example, if the phrase *John took a bus* is accompanied by the phrase *He got off at 10th street*, it is possible to use the relation between RIDE_VEHICLE evoked by *take* and DISEMBARKING evoked by *get off*. However, no information about possible fillers of the roles of the RIDE_VEHICLE frame (living being and vehicle) and the TAKING_TIME frame (activity and time duration) is available from FrameNet itself. Future work on SRL using FrameNet should include learning semantic preferences for frame roles from corpora.[24]

Note that SRL based on weighted abduction dramatically differs from the approach presented in Sec. 8.2.2. *Boxer*, as most of the other systems, assigns semantic roles (as well as VerbNet classes and WordNet senses) **before** reasoning starts. Thus, the knowledge base does not contribute to this process. In contrast, *Mini-TACITUS* disambiguates and assigns semantic roles **during** the reasoning process. It selects those word senses and semantic roles, which better contribute to revealing the overall redundancy of the text fragment.

### 8.3.3 *Paraphrasing of Noun Dependencies*

In the abductive setting for RTE, the text serves as a context, while the hypothesis is interpreted with respect to the text (cf. Sec. 8.3.1). Given a text-hypothesis pair containing a noun-noun construction, which is responsible for the entailment that paraphrasing of a

[23]The recall and precision measures are described in Sec. 8.1.2.1.
[24]See Sec. 3.1.2 for corresponding references.

noun dependency, which provides the best matching between the text and the hypothesis is chosen by the reasoner.[25]

As described in Sec. 5.2.3, a special set of axioms was generated from the Proposition Store (Peñas and Hovy, 2010) for interpretation of dependencies between nouns such as noun compounds and possessives. For example, given the *T-H* pair containing the phrases *Germany's 82 million people* and *82 million people live in Germany*, the following axiom obtained from the Proposition Bank can be applied for drawing the inference:

$$people(e_2,x_2) \wedge live(e_3,x_2,u_1,u_2) \wedge in(e_4,e_3,x_1) \wedge Germany(e_1,x_1) \rightarrow$$
$$Germany(e_1,x_1) \wedge people(e_2,x_2) \wedge poss(e_5,x_1,x_2)$$

In the experiment described in this section, the RTE procedure presented in Sec. 8.3.1 was applied to the selected RTE-2 pairs (see Sec. 8.1.3.1) using the axioms extracted from the Proposition Store in combination with the axioms extracted from WordNet and FrameNet (see sections 5.2.1 and 5.2.2).

The results for the relevant *T-H* pairs from the RTE-2 development and test sets are presented in Tables 8.7 and 8.8, where "RTE-2 ID's" stands for the identification numbers assigned to the corresponding entailment pairs in the RTE-2 datasets. 31 of 54 pairs from the development set and 11 from 39 pairs from the test set were resolved correctly. The sets of entailment pairs, to which paraphrasing axioms have been applied, are listed in Tables 8.9 and 8.10. The comment "axiom applied, no entailment predicted" concerns the cases when paraphrasing was applied, but no entailment was predicated, because the related nouns in *T* and *H* could not be mapped. For example, *Gaza's inhabitant* in *T* was paraphrased as *Gaza has inhabitant*, but the paraphrase could not be mapped to *Israel has inhabitant* in *H*, which is correct.

For 39 pairs from the development and test sets, relevant axioms have been found (see Tables 8.9 and 8.10). In 3 cases (e.g., *T*: *native of Beckemeyer*, *H*: *native resides in Beckemeyer*), no paraphrasing axioms have been found and no entailment was predicated, which is correct. For the pair listed below, a wrong interpretation was produced. The hypothesis *The United Nations produces vehicles* was collapsed into the noun compound *United Nations vehicle*, which was linked to the noun compound in the text.

**T:** *Mahbob Amiri, head of Afghanistan's quick-reaction force, said Monday's bomb in the capital detonated as* ***a United Nations vehicle*** *drove past.*

**H:** ***The United Nations produces vehicles***.

[25]The study described in this section was done in collaboration with Anselmo Peñas. Part of the study was published in (Peñas and Ovchinnikova, 2012).

The major number of pairs (20) were resolved on the basis of `npn` propositions. For 12 of them, `n(of)n` propositions were used, e.g., *paper cost – cost of paper*, and for 4 of them, the general rule relating "of" prepositional phrases with possessive constructions was used, which is not surprising. In addition, 4 `nvpn`, 5 `nvn`, 2 `has_instance`, and 1 `has` propositions have been applied (see Tables 8.9 and 8.10).

It is interesting to note that in some cases (6 pairs) the correct interpretation was selected, because pieces of knowledge extracted from WordNet and FrameNet have been evoked together with the axioms extracted from the Proposition Store. Consider the following entailment pairs:

**T:** *One economic study will not be the basis of Canada's public policy decisions, but Easton's research does conclusively show that there are economic benefits in the* ***legalization of marijuana****.*

**H:** ***Drug legalization*** *has benefits.*

Since *marijuana* entails *drug* in WordNet, the noun compound *drug legalization* could be properly mapped to the prepositional phrase *legalization of marijuana*. For two pairs, an appropriate relation (*arsenal – stockpile*, *agility – function*) was missing in WordNet and FrameNet, therefore paraphrasing did not result in a correct interpretation.

Interestingly, not all applied patterns were the most frequent ones. Axioms were applied to 39 noun-noun pairs (see Tables 8.9 and 8.10). Suppose for each two nouns the set of patterns, which include these nouns in the Proposition Store is ordered according to frequency of the patterns ("Pos." in Tables 8.9 and 8.10). Then, for 11 pairs, the most frequent patterns were applied for construction of the best interpretation. 12 selected propositions have a frequency position below 17. Surprisingly, up to 9 selected propositions have a frequency position exceeding 30. This shows that the proposed method for interpretation of noun dependencies in context using abductive inference gives advantages over a purely frequency-based choice of paraphrases.

For 27 pairs (29%), relevant noun-noun constructions could not be extracted. The most typical problem concerns treatment of complex proper names. A number of 23 pairs (24.7%) could not be resolved, because appropriate paraphrasing axioms were missing.

The overall recall of the interpretation of noun dependencies is 45.1%. Unfortunately, we cannot compare the obtained result with results of any other procedure performing a similar task, because interpretation of noun-noun constructions is usually carried out outside the context or using a closed set of artificial semantic relations (cf. Sec. 8.1.3). Never-

Table 8.7 Paraphrasing of noun dependencies in the RTE-2 development set.

| | **Number of pairs** | **RTE-2 development set ID's** |
|---|---|---|
| Correct interpretation | 31 | 18, 20, 52, 71, 117, 129, 175, 188, 281, 283, 284, 304, 337, 363, 373, 387, 392, 397, 464, 467, 503, 566, 591, 649, 654, 688, 716, 728, 746, 747, 785 |
| Wrong interpretation | 1 | 485 |
| No NN construction extracted | 11 | 79, 82, 147, 151, 202, 262, 269, 414, 477, 727, 768 |
| No axiom found | 11 | 91, 175, 207, 243, 311, 371, 639, 689, 721, 746, 800 |
| Total | 54 | |

Table 8.8 Paraphrasing of noun dependencies in the RTE-2 test set.

| | **Number of pairs** | **RTE-2 test set ID's** |
|---|---|---|
| Correct interpretation | 11 | 27, 73, 157, 177, 291, 351, 449, 512, 533, 685, 767 |
| Wrong interpretation | 0 | |
| No NN construction extracted | 16 | 39, 65, 85, 181, 224, 241, 306, 320, 331, 370, 454, 512, 558, 576, 754, 773 |
| No axiom found | 12 | 18, 98, 160, 218, 256, 277, 357, 440, 636, 671, 673, 752 |
| Total | 39 | |

theless, we hope to have shown that interpreting noun compounds and possessives with the help of weighted abduction and axioms extracted from the Proposition Store is promising for natural language understanding. The proposed approach can be extended to treat other semantic phenomena that involve the recovering of implicit predicates. For example given the phrase *to finish the cigarette*, one could collect all `vvn[finish:v X:v cigarette:n]` patterns from the Proposition Store and see what verbs most frequently occupy the position of `X`. Probably, the verb *to smoke* will be one of the most frequent. Similarly, for such phrases as *a fast car*, one could collect the patterns of the form `nvr[car:n X:v fast:r]`, which could help us to find the paraphrase *a car which drives fast*.

Table 8.9 Paraphrasing axioms used to resolve RTE-2 development set pairs.

| **ID's** | **T** | **H** | **Pattern used** | **Pos.** | **Comment** |
|---|---|---|---|---|---|
| 18, 33 | *Germany's people* | *people live in Germany* | `nvpn` | 12 | |
| 387 | *Shapiro's office* | *Shapiro work in office* | `nvpn` | 31 | |

Continued on next page

Table 8.9 – continued from previous page

| ID's | T | H | Pattern used | Pos. | Comment |
|---|---|---|---|---|---|
| 283 | *Gaza's inhabitant* | *Israel has inhabitant* | `nvn` | 2 | axiom applied, no entailment predicted |
| 392 | *Nicholas Cage's wife, Alice Kim Cage* | *Nicholas Cage married Alice Kim Cage* | `nvn` | 67 | |
| 485 | *a United Nations vehicle* | *The United Nations produces vehicles* | `nvn` | 23 | wrong inference |
| 20 | *population for Missouri* | *Missouri's population* | `npn` | 77 | |
| 117 | *trade ban* | *ban on trade* | `npn` | 1 | |
| 188 | *Gulf of Mexico's output* | *output in the Gulf of Mexico* | `npn` | 3 | |
| 304 | *ivory trade* | *trade in ivory* | `npn` | 2 | |
| 782 | *engine problem* | *trouble with engine* | `npn` | 2 | WN used |
| 71 | *Greenpeace founder* | *founder of Greenpeace* | `n(of)n` | 1 | |
| 129 | *wife of Alexander Solzhenitsyn* | *Alexander Solzhenitsyn's wife* | `n(of)n` | 2 | |
| 284, 566 | *headquarters of WTO* | *WTO headquarters* | `n(of)n` | 3 | |
| 307, 503 | *paper cost* | *cost of paper* | `n(of)n` | 1 | |
| 716 | *paper price* | *cost of paper* | `n(of)n` | 1 | WN used |
| 363 | *legalization of marijuana* | *drugs legalization* | `n(of)n` | 1 | WN used |
| 591, 785 | *legalization of drugs* | *drugs legalization* | `n(of)n` | 1 | |
| 373 | *Amazon share* | *share of Amazon* | `n(of)n` | 1 | |
| 467 | *rabies case* | *case of rabies* | `n(of)n` | 1 | |
| 747 | *barrels of oil* | *oil reserve* | `n(of)n` | 1 | axiom applied, no entailment predicted |
| 281 | *president Akbar Hashemi Rafsanjani* | *Akbar Hashemi Rafsanjani is a president* | `has-instance` | - | |
| 654 | *senator Jader Barbalho* | *Jader Barbalho is a senator* | `has-instance` | - | |
| 52 | *inhabitants of Slovenia* | *Slovenia has inhabitants* | `has` | 1 | |
| 337 | *Basra's governor* | *governor of Basra* | *poss→ of* | - | |

Continued on next page

Table 8.9 – continued from previous page

| ID's | T | H | Pattern used | Pos. | Comment |
|---|---|---|---|---|---|
| 464 | *Boeing's head* | *owner of Boeing* | *poss*→ *of* | - | axiom applied, no entailment predicted |
| 649 | *mistress of Mussolini* | *Mussolini's mistress* | *of*→ *poss* | - | |
| 688 | *board of Regents of the University of California* | *University of California's board of Regents* | *of*→ *poss* | - | |

### 8.3.4 *Domain Text Interpretation*

As described in Sec. 7.3, the proposed NLU reasoning pipeline is able to make use of an ontology formalized using the OWL representation language. We experimented with a toy domain-specific ontology about cars, which is described in Sec. 5.3. Since there are no standard challenges intended for the evaluation of domain-specific NLU, we had to construct a test set from scratch. Building a reliable and representative test set for an NLU evaluation is a complex task itself, which requires a lot of manual work. Such a project is out of the scope of this study. Instead, we provide results of a small experiment on domain-specific text interpretation. The experiment represents a qualitative rather than quantitative study.

The proposed procedure was tested using textual entailment pairs, which were constructed as follows. First, we used the World Wide Web corpus for extracting 50 sentences containing at least two lexemes occurring in our lexicon-ontology mapping (cf. Sec. 5.3). We performed only minimal correction of texts extracted from the web resolving anaphora where necessary. Then, we selected 30 sentences, which seemed to be maximally diverse with respect to vocabulary and structure. Using these sentences as "texts", we manually generated "hypotheses" making sure that the resulting *T*-*H* pairs account for diverse NLU aspects.

Given the *T*-*H* pairs, we asked two experts in the car domain to annotate them with "entailment"/"no entailment" using the RTE-2 guidelines[26] for instructing the annotators. The annotators agreed about 21 pairs out of 30 (13 entailments, 8 non-entailments). The final test set constitutes 8 entailment pairs and 8 non-entailment pairs equally annotated by both experts. The test pairs were processed by *Mini-TACITUS* as described in Sec. 8.3.1.

[26] *http://pascallin.ecs.soton.ac.uk/Challenges/RTE2/Instructions/*

Table 8.10 Paraphrasing axioms used to resolve RTE-2 test set pairs.

| ID's | T | H | Pattern used | Pos. | Comment |
|---|---|---|---|---|---|
| 73 | *First Automotive Works Group, China's maker* | *First Automotive Works Group is based in China* | `nvpn` | 54 | |
| 291 | *Japan's agency* | *agency is based in Japan* | `nvpn` | 82 | |
| 27 | *U.S. Ambassador to Rome Mel Sembler* | *Mel Sembler represents the U.S.* | `nvn` | 46 | |
| 685 | *Microsoft's monopoly* | *Microsoft holds a monopoly* | `nvn` | 7 | |
| 157 | *case against Jackson* | *Jackson trial* | `npn` | 44 | FN used |
| 449 | *Ambassador to Rome* | *Rome's council* | `npn` | 17 | axiom applied, no entailment predicted |
| 767 | *Berlin's landmark* | *landmark in Berlin* | `npn` | 2 | |
| 177 | *price of gasoline* | *gas price* | `n(of)n` | 1 | WN used |
| 351 | *Hamas leader* | *member of Hamas* | `n(of)n` | 2 | WN used |

Since in this experiment there was no training set for defining an entailment predicting threshold, entailment was predicted only if the hypothesis was fully entailed by the text. In the following, we consider the result of processing for each *T*-*H* pair in detail.

(8.1)
**T**: *Every Mercedes has had airbags standard for the past twenty years.*
**H**: *A Mercedes car has safety features.*

Concerning Ex. (8.1), entailment was correctly detected by *Mini-TACITUS* using the hierarchical link between *airbag* and *safety features* from the car ontology. Note that the lexeme *Mercedes* can refer both to the manufacturer called "Mercedes" and to a car with the corresponding brand. It was correspondingly linked to both ontological nodes in the constructed lexicon-ontology mapping.

(8.2)
**T**: *The company has one of the Uk's widest selection of used Porsche Boxster Cabriolet cars for sale.*
**H**: *The company sells convertibles.*

For Ex. (8.2), entailment was not predicted. The lexeme *convertible* in *H* was correctly linked to *cabriolet* in *T* using the ontology, but the paraphrase *has for sale – sells* could not be resolved.

(8.3)
**T**: *The first three tests of the prototype Boxster, equipped with electric motors, will launch in early 2011.*
**H**: *Boxster is an electric car.*

Entailment was correctly predicted for Ex. (8.3) using the definition of *electric car* in the ontology as "having an electric motor".

(8.4)
**T**: *We traded in our awesome Opel Corsa Twinport with upgraded sound set for it.*
**H**: *Corsa Twinport is produced by Opel.*

Example (8.4) is somewhat controversial. The entailment was predicted not because the noun compound *Opel Corsa Twinport* has been properly resolved, but because the ontology already contains the information about Corsa Twinport being produced by Opel. Thus, the hypothesis is entailed by the knowledge base alone.

(8.5)
**T**: *Find Fiat Punto Grande Alloy Wheels with locking wheel nuts in Vehicle Parts Accessories.*
**H**: *Fiat Punto Grande has alloy wheels.*

Example (8.5) is similar to Ex. (8.4). The noun compound *Fiat Punto Grande Alloy Wheels* was not resolved, but the fact about Fiat Punto Grande having alloy wheels is already contained in the ontology.

(8.6)
**T**: *Davis has a new car. The vehicle has 2 alarm systems.*
**H**: *A car has an alarm system.*

Example (8.6) was correctly resolved; *car* was linked to *vehicle*, and the formula $\exists$HAS_ALARMSYSTEM.(ALARM_SYSTEM) evoked by *H* was proven to be entailed by $\exists$HAS_ALARMSYSTEM.(ALARM_SYSTEM)$\sqcap \leq$ 2 HAS_ALARMSYSTEM$\sqcap$
$\geq$ 2 HAS_ALARMSYSTEM evoked by *T*.

(8.7)
**T**: *Audi TT is the most popular German car.*
**H**: *Audi TT is produced in Europe.*

Example (8.7) was not resolved correctly. The proposition $German(e_1, x_1)$ was linked to the concept GERMAN($x_1$). However, the logical form fragment $produce(e_1, x_1, x_2) \wedge in(e_2, e_1, x_3) \wedge Europe(e_3, x_3)$ was not linked to the concept description EUROPEAN($x_2$), because there was no corresponding lexicon-ontology mapping.

(8.8) **T**: *Electric cars in Germany could pollute more than gasoline cars.*
**H**: *Germany has gasoline cars.*

Given Ex. (8.8), entailment was predicted, which corresponds to the expert's annotation. However, the prediction results from a wrong analysis. In the interpretation of *T*, both *electric cars* and *gasoline cars* were set to refer to the same entity (because the *car* propositions were merged), which is wrong. As a consequence, *gasoline cars in Germany* was inferred, which could be paraphrased into *Germany has gasoline cars* using the lexicon-ontology mapping.

(8.9) **T**: *Fiat punto has 3 doors.*
**H**: *Fiat is a 3-door car.*

Example (8.9) represents a non-entailment case. A feature of a specific class (*Fiat punto*) cannot be propagated to a more general class (*Fiat*). However, *Mini-TACITUS* predicts entailment, since *Fiat punto* entails *Fiat*. This is a general problem of inference-based approaches using implicit quantification. Given the following properly quantified logical formulas:

$$\forall x\ (Fiat\_punto(x) \rightarrow \exists y(Door(y) \wedge 3(y) \wedge has(x,y))),$$
$$\forall x\ (Fiat(x) \rightarrow \exists y(Door(y) \wedge 3(y) \wedge has(x,y))),$$
$$\forall x\ (Fiat\_punto(x) \rightarrow Fiat(x)),$$

the incorrect inference does not follow.

(8.10) **T**: *From Porsche also been reported that developing all-electric propulsion technology, including the use of batteries to construct their own electric cars in the future.*
**H**: *Porsche constructs cars run on electric power.*

Example (8.10) does not contain entailment, because *T* does not imply that the Porsche company constructs electric cars (it is only planning to do so). *Mini-TACITUS* is currently unable to handle modal expressions, but the system did not predict entailment, because it failed to detect that *Porsche* is the agent of the *construct* predicate in *T*. Thus, two mistakes have resulted in the correct inference.

(8.11) **T**: *And yet, the largest-selling luxury car in Japan is the Mercedes-Benz, which is made in Germany.*
**H**: *Mercedes is a Japanese car.*

Example (8.11) was correctly resolved by *Mini-TACITUS*. No entailment was predicted, because the concepts *German* and *Japanese* are explicitly disjoint in the ontology.

(8.12) **T**: *Fiat 600 was capable of seating six people within its tiny dimensions.*
**H**: *Fiat 600 has 6 seats.*

Example (8.12) has also been resolved correctly (no entailment), because *H* is inconsistent towards the ontology, which contains the fact about Fiat 600 having exactly four seats.

(13) **T**: *The first air conditioning for cars, was in 1933 when a company in New York city offered installation of air conditioning for cars.*
**H**: *American cars have air conditioning.*

Concerning Ex. (8.13), *H* could not be proven. Thus, no entailment was predicted, which is correct.

(8.14) **T**: *Most interiors are some variation of beige or gray, but the exact color depends on your car's external color.*
**H**: *Car interiors have beige and gray color.*

At present, *Mini-TACITUS* is unable to treat logical connectors such as *and*. No entailment in Ex. (8.14) was predicted, which is correct. However, the proper formula for *H* was not constructed either.

(8.15) **T**: *Most interiors are some variation of beige or a blue/gray, but the exact color depends on your car's external color.*
**H**: *Car's external color is beige.*

Given Ex. (8.15), no entailment was predicted, which is correct.

(8.16) **T**: *There are just 24 buttons on the center console, (remarkably few for a luxury car interior.)*
**H**: *Luxury car interior is made from leather.*

The ontology contains an axiom claiming that all luxury cars have leather interiors. *Mini-TACITUS* has predicted that *H* in Ex. (8.16) is entailed by the knowledge base and the text, although the text has not contributed to the inference. Thus, with respect to the described RTE setting, the prediction of *Mini-TACITUS* was incorrect.

To sum up, the *Mini-TACITUS* system using the car ontology made correct predictions for 12 out of 16 *T-H* pairs. For most of the examples, ontological reasoning was helpful for drawing the appropriate inference. However, 1 positive and 2 negative correct predictions have resulted from a wrong analysis (Ex. (8.8), (8.10), and (8.14)). 3 predictions of which 2 were correct (Ex. (8.4) and (8.5)) and 1 was incorrect (Ex. (8.16)) were made of the basis of the ontology only. The experiment showed some of the already known problems of the proposed approach such as inability to treat logical connectors (Ex. (8.14)) and incompleteness of the non-merge constraints (Ex. (8.8)). Moreover, an interesting problem concerning quantifier-free inference approaches was discovered (Ex. (8.9)), which requires further investigation.

Mapping logical forms to concept descriptions instead of mapping lexemes to concept labels (cf. Sec. 7.3) appeared to be a promising strategy. However, such a mapping requires a lot of careful manual work. Omissions and inaccuracy of the mapping result in lost inferences (cf. Ex. (8.7)). A future work direction concerns automatization of the mapping procedure.

## 8.4 Concluding Remarks

We presented experiments intended for evaluating the proposed approach. Section 8.2 concerns experiments with the *Boxer* and *Nutcracker* systems performed at an early stage of this study. Although these experiments were not particularly successful in terms of performance, they have revealed several interesting features and problems of the deductive approach. First, the inability of a deductive theorem prover to reason given incomplete knowledge made application of the knowledge base almost useless.[27] Instead of a deterministic "yes" or "no" proof provided by a deductive reasoner one would like to have a way of measuring in how far the input formula was proven and which of its parts could not be proven. Second, the semantic role labeling procedure based on the "most frequent sense" strategy and the *SemLink* VerbNet-FrameNet mapping leads to a multiplication of errors and poor labeling as a result.

In Sec. 8.3, we described experiments with the *Mini-TACITUS* system adapted for treating the developed integrative knowledge base. The experiments on recognizing textual entailment (Sec. 8.3.1) and semantic role labeling (Sec. 8.3.2) have shown that the performance of extended *Mini-TACITUS* is compatible with those of the state-of-the-art systems which have been specially designed to solve these tasks. The result of the experiment on paraphrasing of noun dependencies (Sec. 8.3.3) suggests that the abductive approach to paraphrasing noun-noun constructions *in context* is indeed promising. Note that *Mini-*

[27] This corresponds to the observation made by Bos and Markert (2005).

*TACITUS* was not specially adapted to RTE, SRL, or paraphrasing of noun dependencies. Only a little extension was required to make the system able to recognize textual entailment, while SRL and expansion of noun-noun constructions were obtained as a by-product of constructing best interpretations.

Section 8.3.4 describes an experiment on interpreting a domain-specific text using an OWL ontology. This experiment represents a qualitative rather than a quantitative study. It reveals strong and weak sides of the proposed approach and outlines future work directions. Ontological reasoning seem to be well-suited for interpretation of texts from a specific domain. The limitations of this approach mainly concern the difficulties of constructing lexicon-ontology mappings manually. Therefore, future work should include automatization of this process.

The performed experiments show that there is still a lot of space for improving the abductive reasoning procedure. First, for a successful large-scale application of *Mini-TACITUS*, the system needs to be computationally optimized. In its current state, *Mini-TACITUS* requires too much time for producing satisfactory results. As the experiments suggest, speeding up reasoning may lead to significant improvements in system performance. Since *Mini-TACITUS* was not originally designed for large-scale processing, its implementation is in many aspects not efficient enough. Hopefully, it can be improved by changing the data structure and re-implementing some of the main algorithms. Particular attention should be paid to ordering interpretations. At the moment, the system constructs all possible interpretations in order to choose the best one. An "anticipation" mechanism allowing us to construct the most probable interpretations first could allow us to obtain a better interpretation approximation given the time constraints.

Recently, an alternative implementation of weighted abduction based on Integer Linear Programming (ILP) was developed at the Tohoku University (Inoue and Inui, 2011). In this approach, inspired by Santos and Santos (1996), the abductive reasoning problem is formulated as an ILP optimization problem. As it is currently implemented, all possible interpretations are first enumerated. The operations of weighted abduction and the content of axioms are translated into constraints on state variables. Since the goal is to find an interpretation that has a minimum cost, ILP techniques are used to find the assignment of state variables that minimizes the cost function. In a preliminary experiment the ILP-based system achieved a speed-up over *Mini-TACITUS* of two orders of magnitude (Inoue and Inui, 2011).

Second, future work concerns treatment of natural language expressions standing for logical connectors such as implication *if*, negation *not*, disjunction *or*, and others. Quantifiers such as *all*, *each*, *some* as well as operators like *few*, *most*, *many* also require a special account. This advance is needed in order to achieve more precise entailment inferences, which are at the moment based in our approach on the core information content ("aboutness") of texts. In order to achieve this goal, two strategies can be used. First, logical connectors and quantifiers can be axiomatized as proposed by Hobbs (1985b). However, one might doubt that this approach will efficiently work in practice, since the corresponding axioms have relatively high complexity. The second strategy implies providing the abductive reasoner with a new interpretation building algorithm able to construct interpretations for FOL logical forms.

Third, concerning the heuristic non-merge constraints preventing undesired mergings as well as the heuristic for assigning default costs (see Sec. 8.3), a corpus study is needed for evaluating and possibly changing these heuristics. Finding relevant non-merge constraints is an interesting theoretical problem. It is closely related to coreference resolution in natural language, which has always been a challenge for NLP. In the framework of weighted abduction, the problem can be formulated as follows: If $p(x_1,\dots,x_n)$ and $p(y_1,\dots,y_n)$ both hold, how good is that as evidence that $x_i$ and $y_i$ ($i \in \{1,\dots,n\}$) are the same entity, given what else we know about $x_i$ and $y_i$. Different merging and nonmerging constraints can be formulated, which capture different aspects of coreference. Here are some suggestions:

- If $x_i$ and $y_i$ are described by different nouns in the text, they are not the same.
- Two eventualities are not the same unless there is independent evidence that their arguments are the same.
- Propositions that have contradicting properties cannot be unified.
- The likelihood of a merge of $p(x_1,\dots,x_n)$ and $p(y_1,\dots,y_n)$ depends inversely on the frequency of $p$ in the corpus.

These and other heuristics should be investigated in a corpus study and possibly evaluated against an anaphora resolution test set.

## Chapter 9

# Conclusion

Nowadays, the interest in the natural language understanding (NLU) community turns more and more from processing small predefined sets of texts to experimenting with large amounts of data in settings, which are close to situations faced by humans. Great progress has been made, for example, in the field of question answering. Just recently, the question answering system *Watson* developed by IBM researchers managed to win in the quiz show *Jeopardy!* competing against two of the most successful human contestants on the show. Such progress is in large part due to the vast amounts of machine-readable knowledge becoming available (Wikipedia, WordNet, the web document corpus, etc.) and increasing computational capacities of modern computers, which enables us to process this knowledge.

At present, many of the NLP systems successful in practice rely on weakly structured distributional knowledge extracted from corpora with the help of statistical methods. This knowledge is used for measuring semantic similarity between compared text fragments and mapping semantically related fragments to each other. This approach to natural language semantics has obvious limits, because it does not really contribute to understanding of how linguistic knowledge, knowledge about the world, and reasoning mechanisms interact in the process of constructing interpretations of natural language texts.

In contrast to the statistical approaches to natural language semantics, in traditional NLU, a lot of effort has been invested in studying logical properties of natural languages, developing appropriate knowledge representation formats, and reasoning with structured knowledge. The main shortcoming of the existing inference-based approaches is related to the difficulties of applying these approaches in realistic settings, where quick processing of large amounts of "imperfect" text is required. First, inference-based approaches are usually too computationally expensive, which prevents their efficient application to large amounts of data. Second, the bottleneck has always been in obtaining structured knowledge applicable for reasoning. The existence of an appropriate knowledge base has often been taken for granted; the traditional NLU research focused much more on treating logically complex constructions than on equipping the developed systems with enough world knowledge.

A great deal of structured and unstructured world knowledge is nowadays freely available to the community. But how should it be integrated into one knowledge base applicable for reasoning? How should it be made consistent and feasible for NLU? How should it be handled in order to obtain desired inferences? These issues were addressed and investigated in this book.

*Knowledge Base Construction*

We proposed to integrate lexical-semantic, ontological, and distributional knowledge into one modular knowledge base designed for reasoning in the framework of an NLU pipeline. The developed integrative knowledge base contains a) axioms generated using such lexical-semantic resources as WordNet, FrameNet, and the Proposition Store, b) a domain-specific OWL ontology, and c) a similarity space constructed with the help of Latent Semantic Analysis. Building the knowledge base, we did not intend to exploit all existing knowledge resources, because there is definitely much more knowledge available then any NLU system can handle. Our goal was rather to show how some of the large and well-developed knowledge resources can be turned into a knowledge base applicable to natural language understanding and to demonstrate that the proposed strategy has advantages and potential.

One contribution of this part of the research is that we propose new ways of looking at such well-known lexical-semantic resources as WordNet and FrameNet. In the overwhelming majority of NLU systems, WordNet and FrameNet are used for deriving information about synonymy and taxonomic relations. We showed that much more knowledge useful for reasoning can be extracted from these resources: a) non-hierarchical relations in WordNet can be turned into axioms; b) FrameNet syntactic patterns can serve as a good basis for semantic role labeling; c) FrameNet frame relations can be axiomatized and automatically extended. Furthermore, we showed how automatically generated lexical-semantic resources such as PropBank can be turned into structured axioms applicable for reasoning.

Another contribution is that we propose a method, which measures confidence of the axioms using information about corresponding word sense and pattern frequencies in annotated and unannotated corpora. This provides a solution to the problem of defining axiom weights in an abductive framework.

*Ensuring Consistency*

Special attention was paid to the problem of consistency of the developed knowledge base. In this book, we focused on logical and conceptual consistency. To our knowledge, previous studies of conceptual consistency concerned only atomic concepts and taxonomic relations defined on these concepts. In the present study, we considered structured concepts, which are *frames*. Furthermore, we focused on conceptual errors in relations defined on frames, which go beyond taxonomy, such as temporal precedence, causation, perspective, parthood, dependency. These relations were studied using examples from the FrameNet relational network. Ontological constraints were proposed in order to clean up frame relations. Statistical methods were applied for enriching the relational network with more relations and weighting the relations automatically. A case study on cleaning up and expanding the "medical" cluster of frames suggests that applying the proposed methodology can give advantages in NLU. A serious drawback of the developed procedure for conceptual cleaning is that it implies a lot of manual work and a deep understanding of the underlying ontological concepts. Future work in this direction concerns the development of methods for automatic detection and repair of conceptual inconsistencies.

Logical inconsistency was studied with respect to ontologies formalized in Description Logics. A new algorithm for detecting and repairing logical inconsistencies was proposed, which is based on an extended tableau algorithm for Description Logics. The algorithm detects problematic axioms that cause a logical contradiction, distinguishes between different types of logical contradictions, and rewrites the ontology. The algorithm is knowledge preserving in the sense that it keeps as much non-conflicting knowledge contained in the original ontology as possible. The ontology repair procedure was tested using concept definitions in the domain of wine, which were automatically extracted from a domain-specific corpus. In this experiment, a prototype implementation of the algorithm correctly resolved several cases of logical inconsistency. Since distinguishing between some types of logical contradictions cannot always be done automatically, the resulting procedure partially relies on human supervision. In the future, heuristics can be developed in order to automatize this distinction.

*Reasoning with Integrative Knowledge Base*

We experimented with two main forms of inference employed in natural language understanding: deduction and abduction. The natural language understanding system based on deduction, which we used is called *Nutcracker*. The system works in combination with the *Boxer* semantic parser. Performing experiments with *Boxer* and *Nutcracker*, we aimed at investigating whether equipping these systems with additional lexical-semantic knowledge will improve their performance in NLU. We experimented with lexical-semantic knowledge extracted from WordNet and FrameNet. In order to enable *Nutcracker* to access this knowledge, the following steps were performed: 1) adding a word sense disambiguation module based on the *SenseLearner* tool, 2) assigning FrameNet frames and roles to predicate-argument constructions in *Boxer*'s output with the help of the SemLink mapping, 3) equipping *Nutcracker* with a knowledge base consisting of frame relations extracted from FrameNet.

For experimenting with weighted abduction, we employed the abductive reasoner *Mini-TACITUS*. Since the system was not originally designed to process large amounts of data, it was necessary to perform several optimization steps, so that a large knowledge base could be treated. The performed optimization includes 1) introduction of the time and depth parameters forcing the system to terminate the reasoning process and to output the best interpretation constructed so far and 2) filtering out input propositions and axioms, which do not contribute to the construction of the best interpretation. In addition, we provided solutions to two pragmatic problems concerning practical application of weighted abduction to NLU, which are 1) undesirable mergings of propositions having the same predicate name, 2) circular definitions in the knowledge base.

Furthermore, *Mini-TACITUS* was extended in order to make the system able to reason with ontologies and similarity spaces. For reasoning with ontologies, a Description Logic reasoner was integrated into the abductive pipeline. Information about semantic similarity was used in order to link those parts of the discourse, which otherwise could not be linked with the help of the axioms in the knowledge base. Performed optimization steps and extensions enabled successful processing of the selected test sets using the developed integrative knowledge base.

*Evaluation*

We evaluated the proposed approach on three different NLU tasks: recognizing textual entailment (RTE), semantic role labeling (SRL), and paraphrasing of noun dependencies in context. The RTE-2 Challenge dataset was used in performing all three tasks.

Experiments with the *Boxer* and *Nutcracker* systems on RTE and SRL revealed the weak sides of the corresponding approach. First, deductive reasoners – being theoretically and practically well developed – have a significant shortcoming, namely their inability to reason with incomplete knowledge. In the cases when some (even very little) piece of knowledge, relevant for interpretation of a particular text fragment, was missing, *Nutcracker* failed to find a proof. Thus, equipping *Nutcracker* with an extended knowledge base appeared to be useless in practice. Second, the semantic role labeling procedure based on the "most frequent sense" strategy as performed by *Boxer* and on the SemLink mapping from VerbNet to FrameNet resulted in multiplication of errors and a poor result.

In contrast, the performance of the *Mini-TACITUS* system equipped with the developed integrative knowledge base appeared to be compatible with those of the state-of-the-art RTE and SRL systems. The experiment on interpretation of noun dependencies cannot be judged quantitatively because of the small number of the noun-noun constructions relevant for computing entailment in the RTE-2 dataset. However, we believe that the experiment has shown that the proposed strategy of expanding noun-noun constructions with the help of abductive axioms is indeed promising for context-dependent interpretation of noun dependencies. In addition, *Mini-TACITUS* was tested on interpretation of a domain-specific text using a domain-specific ontology. In this experiment, a qualitative rather than a quantitative analysis was performed, which showed some of the interesting features and problems of the weighted abduction approach.

Since the abductive inference procedure and the knowledge base are general and not tuned for a particular task, we consider the results of our experiments to be promising concerning possible manifold applications of weighted abduction in natural language understanding.

The performed experiments suggest that there is still a lot of space for improving the procedure. First, *Mini-TACITUS* algorithms and data structures should be further optimized. Speeding up reasoning may lead to significant improvements in system performance. Second, the heuristics used in the implementation (non-merge constraints, cost function, initial proposition costs) require a careful analysis. Third, the reasoning procedure should be extended to treat natural language expressions standing for logical connectors and quantifiers. This improvement is needed to achieve more precise entailment inferences.

### *Lessons Learned*

Starting this study, we mainly aimed at demonstrating that inference-based NLU has the potential for practical large-scale applications. As mentioned above, two main obstacles to large-scale inference-based NLU have always been 1) lack of structured knowledge applicable for reasoning and 2) computational complexity of reasoning.

Concerning the first issue, we hope to have shown that world knowledge itself is not a bottleneck anymore. Nowadays, many machine readable resources are indeed freely available to the community. The next challenge is to learn how to use them. What we have clearly observed during our work is that all types of knowledge sources – manually constructed electronic dictionaries, ontologies, distributional knowledge extracted from corpora – have great relevance for NLU and can be perfectly used in combination.

After experimenting with both a deductive and an abductive inference machine, we came to the conclusion that weighted abduction currently seems to be more promising for NLU than classical deductive reasoning, because it a) allows us to reason with incomplete knowledge, b) supports probabilistic reasoning, c) exploits discourse redundancy. Elaborating on an existing abductive reasoner, we aimed at extending it in order to enable reasoning with a large heterogeneous KB, which combines different types of knowledge. We consider the developed inference pipeline to produce promising results in diverse NLU tasks.

A major weakness of our approach still concerns the complexity issue. Although the performed optimization steps allowed us to speed up *Mini-TACITUS*, the system still needs too much time for producing satisfactory results. Therefore, the main *Mini-TACITUS* algorithms require a further optimization.

Nevertheless, we hope that the present research can open new ways for inference-based approaches. Our overall conclusion is that integration of different techniques seems to be the best strategy. Neither strict logical inferences nor distributional approaches alone can provide an exhaustive solution to the problem of natural language understanding. But maybe fuzzy probabilistic reasoning based both on hand crafted knowledge resources and on distributional knowledge mined from corpora can do it.

# Appendix A

**Algorithm** Optimized *Mini-TACITUS* reasoning algorithm

**Require:** logical form *LF* of a text fragment, knowledge base *KB*,
depth parameter *D*, cost parameter *C*, time parameter *T*

**Ensure:** best interpretation $I_{best}$ of *LF*

$I_{init} := \{p(x_1,\ldots,x_n,C,0) | p(x_1,\ldots,x_n) \in LF\}$

$I\_set := \{I_{init}\}$

$apply\_inference(I_{init})$

$Cheapest\_I := \{I | I \in I\_set \text{ } and \text{ } \forall I' \in I\_set : cost(I) \leq cost(I')\}$

$Best\_I := \{I | I \in Cheapest\_I \text{ } and$
$\forall I' \in Cheapest\_I : proof\_length(I) \leq proof\_length(I')\}$

**return** $I_{best}$ which is the first element of *Best_I*

**Subroutine** *apply_inference*

**Require:** interpretation *I*

**while** $processing\_time < T$ **do**
    *NM* :=*generate_non-merge*(*I*)
    *I* :=*factor*(*I*,*NM*)
    *apply_axioms*(*I*)
**end while**

**Subroutine** *generate_non-merge*

**Require:** interpretation $I$

**for** $p(e_1, x_1, \ldots, x_n, c_1, P_1), p(e_2, y_1, \ldots, y_n, c_2, P_2) \in I$ **do**
  **if** ($e_1 \neq e_2$ **and** $p$ is not a noun predicate **and**
    $\exists i \in \{1, \ldots, n\}$: ($x_i \neq y_i$ **and**
    $\exists q'(z_1, ..., x_i, ..., z_m, c_3, P_3), q''(u_1, ..., y_i, ..., u_r, c_4, P_4) \in I$)) **then**
    $NM := NM \cup \{e_1 \neq e_2\}$
  **end if**
**end for**
**return** $NM$

**Subroutine** *factor*

**Require:** $I$, $NM$

**for** $p(x_1, \ldots, x_n, c_1, P_1), p(y_1, \ldots, y_n, c_2, P_2) \in I$ **do**
  **if** $\forall i \in \{1, ..., n\} : x_i \neq y_i \notin NM$ **then**
    $I := I \setminus \{p(x_1, \ldots, x_n, c_1, P_1), p(y_1, \ldots, y_n, c_2, P_2)\} \cup$
    $\{p(x_1, \ldots, x_n, min(c_1, c_2), P_1 \cup P_2)\}$
    $\forall i \in \{1, \ldots, n\}$: replace $y_i$ with $x_i$ in $I$
  **end if**
**end for**

**Subroutine** *apply_axioms*

**Require:** $I$

**for** $\alpha \in KB$ **do**
  $LH :=$ propositions on the left hand side of $\alpha$ with weights
  $RH :=$ propositions on the right hand side of $\alpha$
  **for** $PS \subseteq I$ such that
    $(\forall p(x_1,\ldots,x_n,c,d) \in PS : (d < D$ **and** $\exists p(y_1,\ldots,y_n) \in RH$ ) **and**
    $|PS| = |RH|$ **and** $\exists p(x_1,\ldots,x_n,c,d) \in PS$ such that $c > 0)$ **do**
    $Parents := \bigcup\{p|\exists p'(x_1,\ldots,x_n,c,P) \in PS \textit{ and } p \in P\} \cup$
        $\bigcup\{p|p(x_1,\ldots,x_n,c,P) \in PS\}$
    **if** $\forall p(x_1,\ldots,x_n) \in LH : p \notin Parents$ **then**
      $Cost := 0$
      **for** $p(x_1,\ldots,x_n,c,P) \in PS$ **do**
        $Cost := Cost + c$
      **end for**
      **for** $p(x_1,\ldots,x_n,w) \in LH$, where $w$ is weight **do**
        $NewP := NewP \cup \{p(u_1,\ldots,u_n, w \cdot Cost, Parents)\}$, where
        **if** $\exists q(y_1,\ldots,y_m) \in PS, q(z_1,\ldots,z_m) \in RH$ such that
          $\exists i \in \{1,\ldots,n\}, j \in \{1,\ldots,m\} : x_i = z_j$ **then**
          $u_i = y_j$
        **else**
          $u_i$ is a new variable not contained in $I$
        **end if**
      **end for**
      $I_{new} := I \setminus PS \cup \{p(e,x_1,\ldots,x_n,0,P) \in PS\} \cup NewP$
      $I\_set := I\_set \cup \{I_{new}\}$
      $apply\_inference(I_{new})$
    **end if**
  **end for**
**end for**

# Bibliography


Abney, S. and Light, M. (1999). Hiding a semantic hierarchy in a markov model, in *Proceedings of the Workshop on Unsupervised Learning in Natural Language Processing, ACL*, pp. 1–8.

Agirre, E. and Lacalle, O. L. D. (2003). Clustering WordNet word senses, in *Proceedings of the Conference on Recent Advances on Natural Language*, pp. 121–130.

Akhmatova, E. and Mollá, D. (2005). Recognizing textual entailment via atomic propositions, in *Proceedings of the Workshop on Recognizing Textual Entailment*, pp. 385–403.

Allen, J. (1987). *Natural Language Understanding* (Benjamin/Cummings, Menlo Park, CA).

Allen, J. F. (1984). Towards a general theory of action and time, *Artificial Intelligence* **23**, pp. 123–154.

Alonge, A. and Castelli, M. (2003). Encoding information on metaphoric expressions in WordNet-like resources, in *Proceedings of the ACL workshop on Lexicon and figurative language* (ACL, Morristown, NJ, USA), pp. 10–17.

Andreasen, T. and Nilsson, J. F. (2004). Grammatical specification of domain ontologies, *Data and Knowledge Engineering* **48**, pp. 221–230.

Apresjan, J. D. (1973). Regular polysemy, *Linguistics* **142**, pp. 5–32.

Arya, A., Yaligar, V., Prabhu, R. D., Reddy, R. and Acharaya, R. (2010). A Knowledge Based Approach for Recognizing Textual Entailment for Natural Language Inference using Data Mining, *International Journal on Computer Science and Engineering* **2**, pp. 2133–2140.

Asher, N. (1993). *Reference to Abstract Objects in Discourse* (Kluwer Academic Publishers, Dordrecht).

Asher, N. and Lascarides, A. (1998). Bridging, *Journal of Semantics* **15**, 1, pp. 83–113.

Asher, N. and Lascarides, A. (2003). *Logics of Conversation* (Cambridge University Press).

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D. and Patel-Schneider, P. F. (eds.) (2003). *The Description Logic Handbook: Theory, Implementation, and Applications* (Cambridge University Press, NY).

Baader, F. and Sattler, U. (2001). An overview of tableau algorithms for description logics, *Studia Logica* **69**, pp. 5–40.

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B. and Szpektor, I. (2006). The second PASCAL recognizing textual entailment challenge, in *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*, pp. 1–10.

Baroni, M. and Bernardini, S. (2004). BootCaT: Bootstrapping Corpora and Terms from the Web, in *Proceedings of the International Conference on Language Resources and Evaluation*, pp. 1313–1316.

Barr, A. (1980). Natural language understanding, *AI Magazine* **1**, 1, pp. 5–10.

Barwise, J. and Perry, J. (1980). The Situation Underground, *Stanford Working Papers in Semantics* **1**, p. 155.

Basili, R., Cao, D. D., Marocco, P. and Pennacchiotti, M. (2007). Learning selectional preferences for entailment or paraphrasing rules, in *Proceedings of International Conference on Recent Advances in Natural Language Processing* (Borovets, Bulgaria).

Bateman, J. A., Henschel, R. and Rinaldi, F. (1995). Generalized Upper Model 2.0: documentation, Tech. rep., GMD/Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, Germany, URL `http://www.darmstadt.gmd.de/publish/komet/gen-um/newUM.html`.

Baumgartner, P. and Kühn, M. (2000). Abducing Coreference by Model Construction, *Journal of Language and Computation* **1**, 2, pp. 175–190.

Bell, D., Qi, G. and Liu, W. (2007). Approaches to Inconsistency Handling in Description-Logic Based Ontologies, in *OTM Confederated International Workshops and Posters*, LNCS 4806 (Springer), pp. 1303–1311.

Berry, M. W., Drmac, Z. and Jessup, E. R. (1999). Matrices, Vector Spaces, and Information Retrieval, *SIAM Review* **41**, pp. 335–362.

Black, F. (1964). *A Deductive Question Answering System*, Ph.D. thesis, Harvard University.

Blackburn, P. and Bos, J. (2005). *Representation and Inference for Natural Language: A First Course in Computational Semantics* (CSLI Publications).

Blackburn, P., Bos, J., Kohlhase, M. and de Nivelle, H. (2001). Inference and computational semantics, *Computing Meaning* **2**, pp. 11–28.

Bobrow, D., Kaplan, R., Kay, M., Norman, D., Thompson, H. and Winograd, T. (1977). GUS, A Frame-Driven Dialogue System, *Artificial Intelligence* **8**, pp. 155–173.

Bobrow, D. G. (1964). *Natural Language Input for a Computer Problem Solving System*, Ph.D. thesis, Cambridge, MA, USA.

Bolinger, D. (1965). The atomization of meaning, *Language* **41**, pp. 555–573.

Bos, J. (2003). Exploring Model Building for Natural Language Understanding, in P. Blackburn and J. Bos (eds.), *Proceedings of the Workshop on Inference in Computational Semantics*, pp. 41–55.

Bos, J. (2008). Wide-Coverage Semantic Analysis with Boxer, in J. Bos and R. Delmonte (eds.), *Proceedings of the Semantics in Text Processing Conferece*, Research in Computational Semantics (College Publications), pp. 277–286.

Bos, J. (2009). Applying automated deduction to natural language understanding, *Journal of Applied Logic* **7**, 1, pp. 100–112.

Bos, J. and Markert, K. (2005). Recognizing textual entailment with logical inference, in *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pp. 628–635.

Bos, J. and Markert, K. (2006). Recognising Textual Entailment with Robust Logical Inference, in J. Quinonero-Candela, I. Dagan, B. Magnini and F. d'Alché Buc (eds.), *Proceedings of Machine Learning Challenges*, *LNAI*, Vol. 3944, pp. 404–426.

Brandt, S., Küsters, R. and Turhan, A.-Y. (2002). Approximating $\mathscr{ALCN}$-concept descriptions, in *Proceedings of the 2002 International Workshop on Description Logics*.

Bryl, V., Giuliano, C., Serafini, L. and Tymoshenko, K. (2010). Supporting natural language processing with background knowledge: Coreference resolution case, in *Proceedings of the International Semantic Web Conference*, pp. 80–95.

Buitelaar, P. and Cimiano, P. (eds.) (2008). *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, *Frontiers in Artificial Intelligence and Applications*, Vol. 167 (IOS Press, Amsterdam).

Buitelaar, P. and Siegel, M. (2006). Ontology-based Information Extraction with SOBA, in *Proceedings of the International Conference on Language Resources and Evaluation*, pp. 2321–2324.

Bullinaria, J. and Levy, J. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study, *Behavior Research Methods* , 3, pp. 510–526.

Burchardt, A., Erk, K. and Frank, A. (2005). A WordNet Detour to FrameNet, in B. Fisseni, H.-C. Schmitz, B. Schrder and P. Wagner (eds.), *Sprachtechnologie, mobile Kommunikation und linguistische Resourcen* (Lang, Peter, Frankfurt am Main), p. 16.

Burchardt, A. and Pennacchiotti, M. (2008). FATE: a FrameNet-Annotated Corpus for Textual Entailment, in *Proceeding of the International Conference on Language Resources and Evaluation*.

Burchardt, A., Pennacchiotti, M., Thater, S. and Pinkal, M. (2009). Assessing the impact of frame semantics on textual entailment, *Natural Language Engineering* **15**, 4, pp. 527–550.

Butnariu, C., Kim, S. N., Nakov, P., Ó Séaghdha, D., Szpakowicz, S. and Veale, T. (2010). SemEval-2 Task 9: The Interpretation of Noun Compounds Using Paraphrasing Verbs and Prepositions, in *Proceedings of the International Workshop on Semantic Evaluation* (ACL, Uppsala, Sweden), pp. 39–44.

Cao, D. D., Croce, D., Pennacchiotti, M. and Basili, R. (2008). Combining word sense and usage for modeling frame semantics, in *Proceeding of the Semantics in Text Processing Conference*, pp. 85–101.

Carroll, J. and McCarthy, D. (2000). Word sense disambiguation using automatically acquired verbal preferences, *Computers and the Humanities* **34**, pp. 109–114.

Castano, S., Espinosa, S., Ferrara, A., Karkaletsis, V., Kaya, A., Möller, R., Montanelli, S., Petasis, G. and Wessel, M. (2008). Multimedia interpretation for dynamic ontology evolution, in *Journal of Logic and Computation*, Vol. 19 (Oxford University Press), pp. 859–897.

Cederberg, S. and Widdows, D. (2003). Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy, in *Proceeding of the Conference on Natural Language Learning*, pp. 111–118.

Charniak, E. (1972). *Toward a Model of Children's Story Comprehension*, Ph.D. thesis, MIT.

Charniak, E. and Goldman, R. P. (1989). A semantics for probabilistic quantifier-free first-order languages, with particular application to story understanding, in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1074–1079.

Chklovski, T. and Pantel, P. (2004). VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations, in D. Lin and D. Wu (eds.), *Proceedings of the Conference on Empirical Methods on Natural Language Processing* (ACL), pp. 33–40.

Chomsky, N. (1965). *Aspects of the theory of syntax* (M.I.T. Press, Cambridge, Massachusetts).

Chomsky, N. (1976). Conditions on Rules of Grammar, *Linguistic Analysis* **2**, pp. 303–351.

Church, K. W. and Hanks, P. (1989). Word association norms, mutual information, and lexicography, in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 76–83.

Cimiano, P. (2003). Building models for bridges, in *Proceedings of the Workshop on Inference in Computational Semantics*, pp. 57–71.

Cimiano, P., Haase, P., Herold, M., Mantel, M. and Buitelaar, P. (2007). LexOnto: A Model for Ontology Lexicons for Ontology-based NLP, in *Proceeding of the Workshop on Ontologies and Lexical Resources*.

Cimiano, P. and Völker, J. (2005). Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery, in A. Montoyo, R. Munoz and E. Metais (eds.), *Proceedings of the International Conference on Applications of Natural Language to Information Systems*, *LNCS*, Vol. 3513 (Springer), pp. 227–238.

Cimiano, P. and Wenderoth, J. (2007). Automatic Acquisition of Ranked Qualia Structures from the Web, in *Proceeding of the Annual Meeting of the Association for Computational Linguistics*, pp. 888–895.

Claessen, K. and Sörensson, N. (2003). New Techniques that Improve MACE-style Finite Model Finding, in P. Baumgartner and C. G. Fermüller (eds.), *Proceedings of Conference on Automated Deduction*.

Clark, H. H. (1975). Bridging, in R. C. Schank and B. L. Nash-Webber (eds.), *Theoretical issues in natural language processing* (Association for Computing Machinery).

Clark, P. and Harrison, P. (2008). Recognizing Textual Entailment with Logical Inference , in *Proceedings of the Text Analysis Conference* (Gaithsburg, Maryland).

Clark, P., Murray, W. R., Thompson, J., Harrison, P., Hobbs, J. and Fellbaum, C. (2007). On the role of lexical and world knowledge in RTE3, in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing* (ACL, Morristown, NJ, USA), pp. 54–59.

Clark, S. and Weir, D. (2002). Class-based probability estimation using a semantic hierarchy, *Computational Linguistics* **28**, pp. 187–206.

Collins, A. and Quillian, M. (1969). Retrieval time from semantic memory, *Journal of Verbal Learning and Verbal Behavior* **8**, 2, pp. 240–247.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*, Ph.D. thesis, University of Pennsylvania.

Copestake, A. (1992). *The Representation of Lexical Semantic Information*, Ph.D. thesis, University of Sussex.

Craig, J. A., Berezner, S. C., Carney, H. C. and Longyear, C. R. (1966). Deacon, Direct English Access and Control, *Proceedings of the Fall Joint Computer Conference* , p. 365.

Cruse, D. (ed.) (1986). *Lexical Semantics* (Cambridge University Press, Cambridge).

Curtis, J., Cabral, J. and Baxter, D. (2006). On the application of the Cyc ontology to word sense disambiguation, in *Proceedings of the International Florida Artificial Intelligence Research Society Conference*, pp. 652–657.

Dagan, I., Dolan, B., Magnini, B. and Roth, D. (2010). Recognizing textual entailment: Rational, evaluation and approaches - Erratum, *Natural Language Engineering* **16**, 1, p. 105.

Dagan, I. and Glickman, O. (2004). Probabilistic Textual Entailment: Generic Applied Modeling of Language Variability, in *Learning Methods for Text Understanding and Mining* (Grenoble, France).

Dagan, I., Glickman, O. and Magnini, B. (2005). The PASCAL Recognizing Textual Entailment Challenge, in *Machine Learning Challenges*, *LNCS*, Vol. 3944 (Springer), pp. 177–190.

Dahlgren, K., McDowell, J. and Stabler, E. P. (1989). Knowledge representation for commonsense reasoning with text, *Computational Linguistics* **15**, pp. 149–170.

Das, D., Schneider, N., Chen, D. and Smith, N. A. (2010). SEMAFOR 1.0: A probabilistic frame-semantic parser, Technical Report CMU-LTI-10-001, Carnegie Mellon University, Pittsburgh, Pennsylvania, URL `http://www.ark.cs.cmu.edu/SEMAFOR/das+schneider+chen+smith.tr10.pdf`.

Davidson, D. (1967). The Logical Form of Action Sentences, in N. Rescher (ed.), *The Logic of Decision and Action* (University of Pittsburgh Press), pp. 81–120.

Davis, M., Logemann, G. and Loveland, D. W. (1962). A machine program for theorem-proving, *Communications of the ACM* **5**, 7, pp. 394–397.

de Bruijn, J. (2003). Using Ontologies, Tech. rep., DERI, URL `http://www.deri.at/fileadmin/documents/DERI-TR-2003-10-29.pdf`.

Deerwester, S. C., Dumais, S., Landauer, T. K., Furnas, G. W. and Harshman, R. A. (1990). Indexing by Latent Semantic Analysis, *American Society of Information Science* **41**, 6, pp. 391–407.

Delmonte, R., Tonelli, S. and Tripodi, R. (2009). Semantic Processing for Text Entailment with VENSES, in *Proceedings of the PASCAL Recognizing Textual Entailment Challenge* (Gaithersburg, Maryland, USA).

Donini, F. M., Colucci, S., Di Noia, T. and Di Sciascio, E. (2009). A tableaux-based method for computing least common subsumers for expressive description logics, in *Proceedings of the International Joint Conference on Artifical intelligence* (Morgan Kaufmann Publishers Inc.), pp. 739–745.

Downing, P. (1977). On the creation and use of English compound nouns, *Language* **53**, 4, pp. 810–842.

Dowty, D. (1991). Thematic Proto-roles and Argument Selection, *Language* **67**, 3, pp. 547–619.

Erk, K. (2007). A simple, similarity-based model for selectional preferences, in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (ACL, Prague, Czech Republic), pp. 216–223.

Erk, K. and Pado, S. (2006). Shalmaneser - a flexible toolbox for semantic role assignment, in *Proceeding of the International Conference on Language Resources and Evaluation* (Genoa, Italy).

Erk, K., Padó, S. and Padó, U. (2010). A flexible, corpus-driven model of regular and inverse selectional preferences, *Computational Linguistics* **36**, 4, pp. 723–763.

Estival, D., Nowak, C. and Zschorn, A. (2004). Towards Ontology-based Natural Language Processing, in *Proceedings of the 4th Workshop on NLP and XML: RDF/RDFS and OWL in Language Technology* (Markowski), pp. 59–66.

Fass, D. (1997). *Processing Metonymy and Metaphor* (Ablex Publishing, Greenwich, CT).

Fellbaum, C. (1998a). Towards a Representation of Idioms in WordNet, in *Proceedings of the workshop on the Use of WordNet in Natural Language Processing Systems (COLING-ACL)*, pp. 52–57.

Fellbaum, C. (ed.) (1998b). *WordNet: an electronic lexical database* (MIT Press).

Fillmore, C. (1968). The case for case, in E. Bach and R. Harms (eds.), *Universals in Linguistic Theory* (Holt, Rinehart, and Winston, New York).

Fillmore, C. (1976). Frame semantics and the nature of language, in *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, Vol. 280, pp. 20–32.

Firth, J. R. (1957). *Papers in Linguistics 1934-1951* (Longmans, London).

Fowler, A., Hauser, B., Hodges, D., Niles, I., Novischi, A. and Stephan, J. (2005). Applying COGEX to Recognize Textual Entailment, in *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 69–72.

Franconi, E. (2002). Description logics for natural language processing, in F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi and P. F. Patel-Schneider (eds.), *Description Logics Handbook* (Cambridge University Press).

Franconi, E. (2003). The Description Logic Handbook, chap. Natural language processing (Cambridge University Press, New York, NY, USA), pp. 450–461.

Frank, A., Krieger, H.-U., Xu, F., Uszkoreit, H., Crysmann, B., Jörg, B. and Schäfer, U. (2005). Querying Structured Knowledge Sources, in *Proceedings of AAAI-05 Workshop on Question Answering in Restricted Domains*, pp. 10–19.

Gardent, C. and Konrad, K. (2000). Interpreting definites using model generation, *Journal of Language and Computation* **2**, pp. 193–209.

Gardent, C. and Webber, B. L. (2001). Towards the Use of Automated Reasoning in Discourse Disambiguation, *Journal of Logic, Language and Information* **10**, 4, pp. 487–509.

Garoufi, K. (2007). *Towards a Better Understanding of Applied Textual Entailment: Annotation and Evaluation of the RTE-2 Dataset*, Master's thesis, Saarland University.

Garrette, D., Erk, K. and Mooney, R. (2011). Integrating logical representations with probabilistic information using markov logic, in *Proceedings of the International Conference on Computational Semantics* (Oxford, England), pp. 105–114.

Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles, *Computational Linguistics* **28**, 3, pp. 245–288.

Girju, R. (2001). Answer fusion with on-line ontology development, in *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics – Student Research Workshop* (Pittsburgh, PA).

Girju, R. (2009). The syntax and semantics of prepositions in the task of automatic interpretation of nominal phrases and compounds: A cross-linguistic study, *Computational Linguistics* **35**, 2, pp. 185–228.

Girju, R., Badulescu, A. and Moldovan, D. (2003). Learning semantic constraints for the automatic discovery of part-whole relations, in *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics* (ACL, Morristown, NJ, USA), pp. 1–8.

Girju, R., Badulescu, A. and Moldovan, D. (2006). Automatic discovery of part-whole relations, *Computational Linguistics* **32**, pp. 83–135.

Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P. and Yuret, D. (2007). SemEval-2007 Task 04: Classification of Semantic Relations between Nominals, in *Proceedings of the International Workshop on Semantic Evaluations* (ACL, Prague, Czech Republic), pp. 13–18.

Green, C. C. and Raphael, B. (1968). The use of theorem-proving techniques in question-answering systems, in *Proceedings of the ACM national conference* (New York, NY, USA), pp. 169–181.

Grefenstette, G. (1994). Corpus-derived first, second and third-order word affinities, in *Proceedings of Euralex*, pp. 279–290.

Groenendijk, J. and Stokhof, M. (1991). Dynamic predicate logic, *Linguistics and Philosophy* **14**, pp. 39–100.

Grosz, B. J. and Sidner, C. L. (1986). Attention, Intentions, and the Structure of Discourse, *Computational Linguistics* **12**, 3, pp. 175–204.

Gruber, T. (2009). Ontology, in *Encyclopedia of Database Systems*, pp. 1963–1965.

Gruber, T. R. (1993). A translation approach to portable ontology specifications, *Knowledge Acquisition* **5**, 2, pp. 199–220.

Guarino, N. (1998). Formal ontology and information systems, in *Proceeding of the International Conference on Formal Ontologies in Information Systems* (Amsterdam, IOS Press), pp. 3–15.

Guarino, N. and Welty, C. (2004). An Overview of OntoClean, in S. Staab and R. Studer (eds.), *Handbook on Ontologies* (Springer), pp. 151–159.

Guha, R. V. and Lenat, D. B. (1990). Cyc: a mid-term report, *AI Mag.* **11**, 3, pp. 32–59.

Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H. and Sure, Y. (2005). A framework for handling inconsistency in changing ontologies, in *Proceedings of the International Semantic Web Conference*, pp. 353–367.

Hagoort, P., Hald, L., Bastiaansen, M. and Petersson, K. M. (2004). Integration of word meaning and world knowledge in language comprehension, *Science* **304**, pp. 438–441.

Harabagiu, S. M., Miller, G. A. and Moldovan, D. I. (1999). WordNet 2 – a morphologically and semantically enhanced resource, in *Proceedings of the Special Interest Group on the Lexicon*, pp. 1–8.

Harabagiu, S. M., Moldovan, D. I., Clark, C., Bowden, M., Hickl, A. and Wang, P. (2005). Employing Two Question Answering Systems in TREC-2005, in *Proceedings of TREC 2005*.

Harris, Z. (1954). Distributional structure, *Word* **10**, 23, pp. 146–162.

Harris, Z. (1968). *Mathematical Structures of Language* (Wiley, New York).

Hays, D. (1964). Dependency theory: A formalism and some observations, *Language* **40**, pp. 511–525.

Hearst, M. (1998). Automated Discovery of WordNet Relations, in C. Fellbaum (ed.), *WordNet: An Electronic Lexical Database and Some of its Applications* (MIT Press).

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora, in *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, p. 8.

Heim, I. (1982). *The Semantics of Definite and Indefinite Noun Phrases*, Ph.D. thesis, University of Massachusetts, Amherst, Amherst, MA.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L. and Szpakowicz, S. (2010). Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals, in *Proceedings of the International Workshop on Semantic Evaluation* (ACL, Uppsala, Sweden), pp. 33–38.

Heymans, S. and Vermeir, D. (2002). A defeasible ontology language, in R. Meersman and Z. T. et al. (eds.), *Confederated International Conferences: CoopIS, DOA, and ODBASE 2002*, LNCS (Springer), pp. 1033–1046.

Hirst, G. (2004). Ontology and the lexicon, in *Handbook on Ontologies*, pp. 209–230.

Hobbs, J. (2009). *Word Meaning and World Knowledge*, chap. VI (Mouton de Gruyter).

Hobbs, J. R. (1985a). On the coherence and structure of discourse, Tech. rep., CSLI-85-37, Center for the Study of Language and Information, Stanford University, URL `http://www.hf.uio.no/ilos/forskning/prosjekter/sprik/docs/pdf/ocsd.pdf`.

Hobbs, J. R. (1985b). Ontological promiscuity, in *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (Chicago, Illinois), pp. 61–69.

Hobbs, J. R., Stickel, M., Appelt, D. and Martin, P. (1993). Interpretation as abduction, *Artificial Intelligence* **63**, pp. 69–142.

Hofstadter, D. R. (1996). *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought* (Basic Books, Inc., New York, NY, USA).

Hsu, M.-H., Tsai, M.-F. and Chen, H.-H. (2008). Combining WordNet and ConceptNet for automatic query expansion: a learning approach, in *Proceedings of the Asia information retrieval conference* (Springer-Verlag), pp. 213–224.

Iftene, A. and Balahur-Dobrescu, A. (2007). Hypothesis transformation and semantic variability rules used in recognizing textual entailment, in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing* (ACL, Morristown, NJ, USA), pp. 125–130.

Inoue, N. and Inui, K. (2011). ILP-Based Reasoning for Weighted Abduction, in *Proceedings of AAAI Workshop on Plan, Activity and Intent Recognition*.

Jackendoff, R. (1983). *Semantics and Cognition* (Cambridge, Mass.: MIT Press).

Jackendoff, R. (1987). The status of thematic relations in linguistic theory. *Linguistic Inquiriy* **18**, pp. 369–412.

Jackendoff, R. S. (1972). *Semantic interpretation in generative grammar* (MA: The MIT Press, Cambridge).

Johnson-Laird, P. N. (1983). *Mental Models: Toward a Cognitive Science of Language, Inference and Consciousness* (Harvard University Press).

Kalyanpur, A. (2006). *Debugging and repair of OWL ontologies*, Ph.D. thesis, College Park, MD, USA.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, Studies in Linguistics and Philosophy (Kluwer, Dordrecht).

Katz, J. J. and Fodor, J. A. (1963). The structure of a Semantic Theory, *Language* **39**, pp. 170–210.

Katz, Y. and Parsia, B. (2005). Towards a nonmonotonic extension to OWL, in *Proceedings of the Workshop OWL: Experiences and Directions*.

Kilgarriff, A. (1997). I don't believe in word senses, in *Computers and the Humanities*, Vol. 31, pp. 91–113.

Kilgarriff, A. (2001). Generative Lexicon Meets Corpus Data: The Case of Nonstandard Word Uses, in P. Bouillon and F. Busa (eds.), *The Language of Word Meaning* (Cambridge University Press, Cambridge), pp. 312–328.

Kipper, K., Dang, H. T. and Palmer, M. (2000). Class-based construction of a verb lexicon, in *Proceedings of the National Conference on Artificial Intelligence and Conference on Innovative Applications of Artificial Intelligence* (AAAI Press / The MIT Press), pp. 691–696.

Kuropka, D. and Becker, J. (2003). Topic-based vector space model, in *Proceedings of the International Conference on Business Information Systems* (Colorado Springs), pp. 7–12.

Lakoff, G. (1987). *Women, Fire and Dangerous Things: What Categories Reveal About the Mind* (University of Chicago Press, Chicago).

Lam, S. C., Pan, J. Z., Sleeman, D. H. and Vasconcelos, W. W. (2006). A Fine-Grained Approach to Resolving Unsatisfiable Ontologies, in *Proceedings of the International Conference on Web Intelligence*, pp. 428–434.

Landauer, T. (2007). *Handbook of Latent Semantic Analysis* (Lawrence Erlbaum Associates).

Landauer, T., Foltz, P. W. and Laham, D. (1998). An Introduction to Latent Semantic Analysis, *Discourse Processes* **25**, pp. 259–284.

Landauer, T. K. and Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge, *Psychological Review* **104**, 2, pp. 211–240.

Langacker, R. W. (1987). *Foundations of cognitive grammar: Theoretical Prerequisites* (Stanford University Press, Stanford, CA).

Lapata, M. and Keller, F. (2004). The web as a baseline: Evaluating the performance of unsupervised web-based models for a range of NLP tasks, in *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 121–128.

Lapata, M. and Lascarides, A. (2003a). Detecting Novel Compounds: The Role of Distributional Evidence, in *Proceedings of the European Chapter of the Association for Computational Linguistics*, pp. 235–242.

Lapata, M. and Lascarides, A. (2003b). A probabilistic account of logical metonymy, *Computational Linguistics* **29**, 2, pp. 263–317.

Lascarides, A. and Asher, N. (1993). Temporal interpretation, discourse relations and commonsense entailment, *Linguistics and Philosophy* **16**, pp. 437–493.

Lauer, M. (1995). *Designing Statistical Language Learners: Experiments on Noun Compounds*, Ph.D. thesis, Macquarie University, Australia.

Leacock, C. and Ravin, Y. (eds.) (2000). *Polysemy: Theoretical and Computational Approaches* (Oxford University Press, New York).

Lenat, D., Miller, G. and Yokoi, T. (1995). CYC, WordNet, and EDR: Critiques and responses, *Communications of the ACM* **38**, 11, pp. 45–48.

Lenci, A., Montemagni, S. and Pirrelli, V. (2006). Acquiring and Representing Meaning. Theoretical and Computational Perspectives, in A. Lenci, S. Montemagni and V. Pirrelli (eds.), *Acquisition and Representation of Word Meaning. Theoretical and Computational Perspectives* (Pisa/Roma, Istituti Editoriali e Poligrafici Internazionali), pp. 19–66.

Levi, J. N. (1978). *The Syntax and Semantics of Complex Nominals* (Academic Press, New York).

Levin, B. (1993). *English Verb Classes and Alternations* (University of Chicago Press, Chicago).

Lin, D. and Pantel, P. (2001). Discovery of Inference Rules for Question Answering, *Natural Language Engineering* **7**, pp. 343–360.

Liu, H. and Singh, P. (2004). ConceptNet – A Practical Commonsense Reasoning Tool-Kit, *BT Technology Journal* **22**, pp. 211–226.

Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence, *Behavior Research Methods Instruments and Computers* **28**, 2, pp. 203–208.

Lund, K., Burgess, C. and Atchley, R. A. (1995). Semantic and associative priming in high-dimensional semantic space, in *Proceedings of the Cognitive Science Society* (Erlbaum Publishers, Hillsdale, NJ, USA), pp. 660–665.

Lyons, J. (1968). *Introduction to Theoretical Linguistics* (University Press, Cambridge).

Lyons, J. (1977). *Semantics, Vol. I & II* (University Press, Cambridge).

Mann, W. and Thompson, S. (1988). Rhetorical Structure Theory: Toward a Functional Theory of text Organization, *International Pragmatics Association Papers in Pragmatics* **8**, 3, pp. 243–281.

Manning, C. D. and Schtze, H. (1999). *Foundations of Statistical Natural Language Processing* (MIT Press, Cambridge, MA).

Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K. and Schasberger, B. (1994). The Penn Treebank: Annotating Predicate-Argument Structure, in *Proceedings of the workshop on Human Language Technology* (ACL), pp. 114–119.

Margolis, E. and Laurence, S. (2005). Concepts, in *Stanford Encyclopedia of Philosophy*.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N. and Oltramari, A. (2003). WonderWeb deliverable D18. the WonderWeb library of foundational ontologies and the DOLCE ontology, Tech. rep., ISTC-CNR, URL `http://wonderweb.semanticweb.org/deliverables/documents/D18.pdf`.

May, R. (1977). *The Grammar of Quantification*, Ph.D. thesis, Massachusetts Institute of Technology.

McCawley, J. D. (1973). *Grammar and Meaning: Papers on Syntactic and Semantic Topics* (Tokyo: Taishukan).

McCord, M. C. (1990). Slot grammar: A system for simpler construction of practical natural language grammars, in *Natural Language and Logic: International Scientific Symposium*, LNCS (Springer), pp. 118–145.

McCord, M. C. (2010). Using Slot Grammar, Tech. rep., IBM T. J. Watson Research Center, RC 23978Revised.

Mccune, W. (2003). Anl/mcs-tm-264 mace4 reference manual and guide, Tech. rep., Argonne National Laboratory, URL `http://www.mcs.anl.gov/uploads/cels/papers/TM-264.pdf`.

Mcdonald, S. and Ramscar, M. (2001). Testing the distributional hypothesis: The influence of context on judgements of semantic similarity, in *Proceedings of the Annual Conference of the Cognitive Science Society*, pp. 611–6.

McGuinness, D. L. and van Harmelen, F. (2004). OWL Web Ontology Language Overview, W3c recommendation, World Wide Web Consortium, URL `http://www.w3.org/TR/2004/REC-owl-features-20040210/`.

Mehdad, Y., Negri, M., Cabrio, E., Kouylekov, M. and Magnini, B. (2009). EDITS: An Open Source Framework for Recognizing Textual Entailment, in *Proceedings of the Text Analysis Conference* (Gaithsburg, Maryland).

Mihalcea, R. and Faruque, E. (2004). SenseLearner: Minimally supervised word sense disambiguation for all words in open text, in *Proceedings of the Annual Meeting of the Association for Computational Linguistics / SIGLEX Senseval-3*, pp. 155–158.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K. (1990). WordNet: An on-line lexical database, *International Journal of Lexicography* **3**, pp. 235–244.

Miller, G. A. and Fellbaum, C. (1991). Semantic networks of English, *Cognition* **41**, 1-3, pp. 197–229.

Minsky, M. (1975). A framework for representing knowledge, in P. Winston (ed.), *The Psychology of Computer Vision* (McGraw-Hill, New York), pp. 211–277.

Mitkov, R. (1999). Anaphora resolution: the state of the art, Tech. rep., University of Wolverhampton, URL `http://clg.wlv.ac.uk/papers/mitkov-99a.pdf`.

Mollá, D. and Vicedo, J. L. (2007). Question answering in restricted domains: An overview, *Computational Linguistics* **33**, pp. 41–61.

Monk, J. D. (1994). *Mathematical Logic* (Springer, New York).

Montague, R. (1973). The proper treatment of quantification in ordinary English, in K. J. J. Hintikka, J. Moravcsic and P. Suppes (eds.), *Approaches to Natural Language* (Reidel, Dordrecht), pp. 221–242.

Morales, L. P., Esteban, A. D. and Gervás, P. (2008). Concept-graph based biomedical automatic summarization using ontologies, in *Proceedings of the 3rd Textgraphs Workshop on Graph-Based Algorithms for Natural Language Processing*, TextGraphs '08 (ACL, Morristown, NJ, USA), ISBN 978-1-905593-57-6, pp. 53–56.

Morato, J., Marzal, M. N., Llorns, J. and Moreiro, J. (2004). Wordnet applications, in *Proceeding of the Global Wordnet Conference* (Brno, Czech Republic).

Motik, B. (2006). *Reasoning in Description Logics using Resolution and Deductive Databases*, Ph.D. thesis, Universitt Karlsruhe (TH), Karlsruhe, Germany.

Motik, B., Shearer, R. and Horrocks, I. (2009). Hypertableau Reasoning for Description Logics, *Journal of Artificial Intelligence Research* **36**, pp. 165–228.

Mulkar, R., Hobbs, J. R. and Hovy, E. (2007). Learning from Reading Syntactically Complex Biology Texts, in *Proceedings of the International Symposium on Logical Formalizations of Commonsense Reasoning* (Palo Alto).

Murphy, G. L. (2002). *The Big Book of Concepts* (MIT Press, Boston, Mass.).

Murphy, L. (2010). *Lexical Meaning* (University Press, Cambrige).

Nakov, P. and Hearst, M. (2006). Using verbs to characterise noun-noun relations, in *Proceedings of the International Conference on Artificial Intelligence: Methodology, Systems and Applications*.

Nastase, V. and Strube, M. (2009). Combining collocations, lexical and encyclopedic knowledge for metonymy resolution, in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Vol. 2 (ACL, Morristown, NJ, USA), pp. 910–918.

Niles, I. and Pease, A. (2001). Towards a standard upper ontology, in *Proceedings of the International Conference on Formal Ontologies in Information Systems* (ACM, NY), pp. 2–9.

Niles, I., Pease, A. and (presenter, A. P. (2003). Linking lexicons and ontologies: Mapping wordnet to the suggested upper merged ontology, in *Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE 03), Las Vegas*, pp. 23–26.

Nirenburg, S. and Raskin, V. (2004). *Ontological Semantics (Language, Speech, and Communication)* (The MIT Press).

Norvig, P. (1983). Frame activated inferences in a story understanding program, in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 624–626.

Norvig, P. (1987). Inference in text understanding, in *Proceedings of National Conference on Artificial Intelligence*, pp. 561–565.

Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Sintek, M., Kiesel, M., Mougouie, B., Vembu, S. B. S., Romanelli, M., Buitelaar, P., Engel, R., Sonntag, D., Reithinger, N., Loos, B., Zorn, H.-P., Micelli, V., Porzel, R., Schmidt, C., Weiten, M., Burkhardt, F. and Zhou, J. (2007). DOLCE ergo SUMO: On Foundational and Domain Models in SWIntO (SmartWeb Integrated Ontology), *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* **5**, 3, pp. 156–174, aCM.

Oltramari, A., Gangemi, A., Guarino, N. and Masolo, C. (2002). Restructuring WordNet's top-level: The OntoClean approach, in *Proceeding of the Workshop on Ontologies and Lexical Resources*, pp. 17–26.

Ovchinnikova, E. and Kühnberger, K.-U. (2006). Adaptive ALE-TBox for extending terminological knowledge, in *Proceedings of the Australian Joint Conference on Artificial Intelligence*, pp. 1111–1115.

Ovchinnikova, E. and Kühnberger, K.-U. (2007). Automatic ontology extension: Resolving inconsistencies, *GLDV-Journal for Computational Linguistics and Language Technology* **22(2)**, pp. 19–33.

Ovchinnikova, E., Montazeri, N., Alexandrov, T., Hobbs, J. R., McCord, M. C. and Mulkar-Mehta, R. (2011). Abductive Reasoning with a Large Knowledge Base for Discourse Processing, in *Proceedings of the International Conference on Computational Semantics* (Oxford, UK), pp. 225–234.

Ovchinnikova, E., Vieu, L., Oltramari, A., Borgo, S. and Alexandrov, T. (2010). Data-Driven and Ontological Analysis of FrameNet for Natural Language Reasoning, in N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner and D. Tapias (eds.), *Proceedings* *of the International Conference on Language Resources and Evaluation* (European Language Resources Association, Valletta, Malta).

Ovchinnikova, E., Wandmacher, T. and Kühnberger, K.-U. (2007). Solving Terminological Inconsistency Problems in Ontology Design, *International Journal of Interoperability in Business Information Systems (IBIS)* **2(1)**, pp. 65–80.

Palmer, M., Gildea, D. and Kingsbury, P. (2005). The Proposition Bank: A Corpus Annotated with Semantic Roles, *Computational Linguistics Journal* **31**, 1, pp. 71–106.

Palmer, M., Ng, H. T. and Dang, H. T. (2006). Evaluation of WSD Systems, in *Word Sense Disambiguation: Algorithms and Applications*, *Text, Speech and Language Technology*, Vol. 33 (Springer, Dordrecht, The Netherlands), pp. 75–106.

Pantel, P., Bhagat, R., Chklovski, T. and Hovy, E. (2007). ISP: Learning inferential selectional preferences, in *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 564–571.

Pantel, P. and Lin, D. (2002). Discovering word senses from text, in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02 (ACM, New York, NY, USA), pp. 613–619.

Papadimitriou, C. M. (1994). *Computational complexity* (Addison-Wesley, Reading, Massachusetts), ISBN 0201530821.

Partee, B. and Borschev, V. (1998). Integrating lexical and formal semantics: Genitives, relational nouns, and type-shifting, in *Proceedings of Second Tbilisi Symposium on Language, Logic, and Computation* (Tbilisi), pp. 229–241.

Peñas, A. and Hovy, E. (2010). Filling knowledge gaps in text for machine reading, in *Proceedings of Coling 2010: Posters* (Coling 2010 Organizing Committee, Beijing, China), pp. 979–987.

Peñas, A. and Ovchinnikova, E. (2012). Unsupervised acquisition of axioms to paraphrase noun compounds and genitives, in *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, LNCS (Springer, New Delhi, India).

Pease, A., Sutcliffe, G., Siegel, N. and Trac, S. (2008). The annual sumo reasoning prizes at casc. in B. Konev, R. A. Schmidt and S. Schulz (eds.), *PAAR/ESHOL*, *CEUR Workshop Proceedings*, Vol. 373 (CEUR-WS.org).

Pennacchiotti, M. and Wirdth, M. (2009). Measuring Frame Relatedness, in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 657–665.

Peraldi, S. E., Kaya, A., Melzer, S. and Möller, R. (2008). On ontology based abduction for text interpretation, in A. Gelbukh (ed.), *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, no. 4919 in LNCS (Springer), pp. 194–205.

Pereira, F., Tishby, N. and Lee, L. (1993). Distributional clustering of english words, in *Proceedings of the Annual Meeting on Association for Computational Linguistics* (ACL, Morristown, NJ, USA), pp. 183–190.

Pianesi, F. and Varzi, A. (2000). Events and event talk: An introduction, in *Speaking of Events* (Oxford University Press), pp. 3–47.

Poesio, M. (2005). Domain modelling and NLP: Formal ontologies? Lexica? Or a bit of both? *Applied Ontology* **1**, 1, pp. 27–33.

Prevot, L., Borgo, S. and Oltramari, A. (2005). Interfacing Ontologies and Lexical Resources, in *Proceedings of the Workshop on Ontologies and Lexical Resources* (Gaithersburg, Maryland, USA), pp. 1–12.

Prevot, L., Borgo, S. and Oltramari, A. (2009). Interfacing Ontologies and Lexical Resources, *Ontologies and the Lexicon* , pp. 195–216.

Prevot, L., Huang, C.-R., Calzolari, N., Gangemi, A., Lenci, A. and Oltramari, A. (2010). *Ontology and the lexicon: a multidisciplinary perspective* (University Press, Cambrige), pp. 3–24.

Pustejovsky, J. (1991). The Generative Lexicon, *Computational Linguistics* **17**, 4, pp. 409–441.

Pustejovsky, J. (1995). *The Generative Lexicon* (The MIT Press, Cambridge).

Pustejovsky, J., Havasi, C., Littman, J., Sauri, R., Rumshisky, A. and Verhagen, M. (2006). Towards a Generative Lexical Resource: The Brandeis Semantic Ontology, in *Proceeding of the International Conference on Language Resources and Evaluation*.

Putnam, H. (1975). *Mind, Language and Reality: Philosophical Papers, Volume 2* (Cambridge University Press).

Quillian, M. R. (1968). Semantic memory, *Semantic Information Processing* , pp. 227–270.

Raphael, B. (1964). Sir: a computer program for semantic information retrieval, Tech. rep., Cambridge, MA, USA, URL `http://www.bitsavers.org/pdf/mit/ai/aim/AITR-220.pdf`.

Reed, S. L. and Lenat, D. B. (2002). Mapping Ontologies into Cyc, Tech. rep., Cycorp, URL `http://www.cyc.com/doc/white_papers/mapping-ontologies-into-cyc_v31.pdf`.

Resnik, P. (1997). Selectional Preference and Sense Disambiguation, in *ACL SIGLEX Workshop on Tagging Text with Lexical Semantics*, pp. 52–57.

Riazanov, A. and Voronkov, A. (2002). The design and implementation of vampire, *AI Communications* **15**, pp. 91–110.

Richardson, M. and Domingos, P. (2006). Markov logic networks, *Machine Learning* **62**, 1-2, pp. 107–136.

Richens, T. (2008). Anomalies in the WordNet verb hierarchy, in *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08 (ACL, Morristown, NJ, USA), ISBN 978-1-905593-44-6, pp. 729–736.

Riemer, N. (2010). *Introducing semantics* (Cambridge University Press).

Robinson, J. A. (1965). A machine-oriented logic based on the resolution principle, *J. ACM* **12**, pp. 23–41.

Robinson, J. A. and Voronkov, A. (eds.) (2001). *Handbook of Automated Reasoning* (Elsevier and MIT Press).

Rooth, M., Riezler, S., Prescher, D., Carroll, G. and Beil, F. (1999). Inducing a semantically annotated lexicon via em-based clustering, in *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (Maryland, MD).

Rosch, E. (1975). Cognitive representations of semantic categories, *Journal of Experimental Psychology: General* **104**, 3, pp. 192–233.

Rosch, E. (1978). *Principles of Categorization* (Lawrence Erlbaum Associates, Hillsdale (NJ), USA), pp. 27–48.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy, *Communications of the ACM* **8**, pp. 627–633.

Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C. and Scheffczyk, J. (2010). FrameNet II: Extended Theory and Practice, Tech. rep., Berkeley, USA, URL `http://www.cs.pitt.edu/~wiebe/courses/CS2731/Fall06/frameNetBook1.3.pdf`.

Russell, S. J., Norvig, P., Candy, J. F., Malik, J. M. and Edwards, D. D. (1995). *Artificial intelligence: a modern approach* (Prentice-Hall, Inc., Upper Saddle River, NJ, USA).

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval* (McGraw-Hill, Inc., New York, NY, USA).

Santos, E. and Santos, E. S. (1996). Polynomial solvability of cost-based abduction, *Artificial Intelligence* **86**, pp. 157–170.

Schank, R. (1975). *Conceptual Information Processing*, *Fundamental Studies in Computer Science*, Vol. 3 (North-Holland, Amsterdam).

Schank, R. and Abelson, R. (1977). *Scripts, plans, goals and understanding: An inquiry into human knowledge structures* (Lawrence Erlbaum Associates, Hillsdale, NJ.).

Schank, R. and Cleary, C. (1995). *Engines for education* (Erlbaum Assoc, Hillsdale, NJ).

Schank, R. C. (1972). Conceptual Dependency: A Theory of Natural Language Understanding, *Cognitive Psychology* **3**, 4, pp. 532–631.

Schank, R. C. (1991). *Tell Me a Story: A New Look at Real and Artificial Intelligence* (Simon & Schuster, New York).

Scheffczyk, J., Baker, C. F. and Narayanan, S. (2006). Ontology-based reasoning about lexical resources, in *Proceeding of the Workshop on Ontologies and Lexical Resources*.

Schlobach, S. and Cornet, R. (2003). Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies, in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 355–362.

Schulte im Walde, S. (2010). Comparing Computational Approaches to Selectional Preferences – Second-Order Co-Occurrence vs. Latent Semantic Clusters, in *Proceedings of the International Conference on Language Resources and Evaluation* (Valletta, Malta), pp. 1381–1388.

Schütze, H. (1998). Automatic Word Sense Discrimination, *Computational Linguistics* **24**, 1, pp. 97–123.

Shen, D. and Lapata, M. (2007). Using Semantic Roles to Improve Question Answering, in *Proceeding of the Joint Meeting of the Conference on Empirical Methods on Natural Language Processing and the Conference on Natural Learning*, pp. 12–21.

Shieber, S. M. (1986). *An Introduction to Unification-Based Approaches to Grammar*, *CSLI Lecture Notes Series*, Vol. 4 (Center for the Study of Language and Information, Stanford, CA).

Shnarch, E., Barak, L. and Dagan, I. (2009). Extracting lexical reference rules from wikipedia, in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Vol. 1 (ACL, Morristown, NJ, USA), pp. 450–458.

Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T. and Zhu, W. L. (2002). Open mind common sense: Knowledge acquisition from the general public, in *Proceedings of ODBASE02* (Springer-Verlag), pp. 1223–1237.

Smith, B. (1996). Mereotopology: A Theory of Parts and Boundaries, *Data and Knowledge Engineering* **20**, 3, pp. 287–303.

Smith, B. (2004). Beyond concepts: Ontology as reality representation, in *Proceedings of the International Conference on Formal Ontologies in Information Systems* (IOS Press), pp. 73–84.

Sowa, J. (2000). *Knowledge Representation: Logical, Philosophical and Computational Foundations* (Brooks/Cole, Pacific Grove, CA).

Sowa, J. F. (1987). Semantic Networks, *Encyclopedia of Artificial Intelligence* .

Steedman, M. (2001). *The syntactic process* (MIT Press, Cambridge, MA, USA).

Stickel, M. E. (1988). A prolog technology theorem prover: Implementation by an extended prolog compiler, *Journal of Automated Reasoning* **4**, 4, pp. 353–380.

Stickel, M. E. (1990). Rationale and methods for abductive reasoning in natural-language interpretation, in Studer (ed.), *Natural Language and Logic*, *LNCS*, Vol. 459 (Springer), pp. 233–252.

Suchanek, F. M. (2008). *Automated Reasoning and Common Sense* (VDM).

Suchanek, F. M., Kasneci, G. and Weikum, G. (2007). Yago: A Core of Semantic Knowledge, in *Proceedings of the international World Wide Web conference* (ACM Press, New York, NY, USA).

Talmy, L. (1983). How language structures space, in H. Pick and L. Acredolo (eds.), *Spatial Orientation: Theory, Research, and Application*, pp. 225–282.

Talmy, L. (2000). *Toward a Cognitive Semantics* (MIT Press, Cambridge, MA).

Tanenhaus, M. K. and Brown-Schmidt, S. (2008). Language processing in the natural world, *Philosophical Transactions of the Royal Society B: Biological Sciences* **363**, pp. 1105–1122.

Tarnawsky, Y. (1982). *Knowledge semantics*, Ph.D. thesis, New York, USA.

Tatu, M., Iles, B., Slavick, J., Novischi, A. and Moldovan, D. (2006). COGEX at the second recognizing textual entailment challenge, in *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*.

Tatu, M. and Moldovan, D. (2005). A semantic approach to recognizing textual entailment, in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* (ACL, Morristown, NJ, USA), pp. 371–378.

Tratz, S. and Hovy, E. (2010). A taxonomy, dataset, and classifier for automatic noun compound interpretation, in *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (ACL, Uppsala, Sweden), pp. 678–687.

Tsatsaronis, G. and Panagiotopoulou, V. (2009). A generalized vector space model for text retrieval based on semantic relatedness, in *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop* (ACL, Morristown, NJ, USA), pp. 70–78.

Velardi, P., Navigli, R., Cucchiarelli, A. and Neri, F. (2006). Evaluation of OntoLearn, a methodology for automatic population of domain ontologies, in P. Buitelaar, P. Cimiano and B. Magnini (eds.), *Ontology Learning from Text: Methods, Applications and Evaluation* (IOS Press).

Verdezoto, N. and Vieu, L. (2011). Towards semi-automatic methods for improving WordNet, in J. Bos and S. Pulman (eds.), *Proceedings of the Ninth International Conference on Computational Semantics* (Oxford, UK), pp. 275–284.

Vermazen, B. (1967). Review of Jerrold Katz and Paul Postal, An Integrated Theory of Linguistic Descriptions, and Katz, Philosophy of Language, *Synthese* **17**, 1, pp. 350–365.

Viegas, E., Beale, S. and Nirenburg, S. (1998). The computational lexical semantics of syntagmatic relations, in *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Conference on Computational Linguistics*, pp. 1328–1332.

Vieu, L. (2009). *Representing Content - Semantics, Ontology, and their Interplay*, Habilitation diriger des recherches, Universit Paul Sabatier, Toulouse, France.

Völker, J., Hitzler, P. and Cimiano, P. (2007). Acquisition of OWL DL Axioms from Lexical Resources, in *Proceedings of the 4th European conference on The Semantic Web: Research and Applications*, ESWC '07 (Springer-Verlag, Berlin, Heidelberg), ISBN 978-3-540-72666-1, pp. 670–685.

Völker, J., Vrandečić, D., Sure, Y. and Hotho, A. (2008). AEON - An approach to the automatic evaluation of ontologies, *Applied Ontology* **3**, 1-2, pp. 41–62.

Vossen, P. (2002). Eurowordnet general document, Tech. rep., Amsterdam: Vrije Universiteit, URL `http://www.hum.uva.nl/~ewn`.

Wandmacher, T. (2005). How semantic is Latent Semantic Analysis? in *Proceeding of the TALN/RECITAL Conference* (Dourdan, France).

Wandmacher, T. (2008). *Adaptive word prediction and its application in an assistive communication system*, Ph.D. thesis, University of Tübingen, Germany.

Wandmacher, T., Ovchinnikova, E. and Alexandrov, T. (2008). Does Latent Semantic Analysis reflect Human Associations? in *Proceeding of the Lexical Semantics workshop at ESSLLI'08* (Hamburg, Germany).

Wang, H., Horridge, M., Rector, A. L., Drummond, N. and Seidenberg, J. (2005). Debugging OWL-DL ontologies: A heuristic approach. in *Proceedings of the International Semantic Web Conference*, pp. 745–757.

Warren, B. (1978). *Semantic Patterns of Noun-Noun Compounds* (Acta Universitatis Gothoburgensis, Göteborg).

Weizenbaum, J. (1966). ELIZA – A computer program for the study of natural language communication between man and machine, *Communications of the ACM* **9**, 1, pp. 36–45.

Wilensky, R. (1983). *Planning and Understanding: A Computational Approach to Human Reasoning* (Addison-Wesley, Reading, MA).

Wilensky, R., Chin, D. N., Luria, M., Martin, J. H., Mayfield, J. and Wu, D. (1988). The Berkeley UNIX Consultant Project, *Computational Linguistics* **14**, 4, pp. 35–84.

Wilks, Y. (2002). Ontotherapy: or, how to stop worrying about what there is, in *Proceedings of the Workshop on Ontologies and Lexical Resources*.

Winograd, T. (1972). *Understanding Natural Language* (Academic Press, Inc., Orlando, FL, USA).

Winston, M. E., Chaffin, R. and Herrmann, D. (1987). A taxonomy of part-whole relations, *Cognitive Science* **11**, pp. 417–444.

Woods, W. A., Bates, M., Brachman, R., Bobrow, R., Cohen, P., Goodman, B., Israel, D., Schmolze, J. and Sidner, C. (1980). Research in knowledge representation for natural language understandingannual report (9/1/79- 8/31/80), Bbn report 4513, World Wide Web Consortium, Belt Boranek and Newman, Cambridge, MA.

Woods, W. A., Kaplan, R. and Webber, N. B. (1972). The LUNAR sciences natural language information system: Final report, Tech. Rep. BBN Report No. 2378, Bolt Beranek and Newman, Cambridge, Massachusetts.

# Index


**A**

algorithms
  Mini-TACITUS algorithm, 156, 157
  ontology rewriting algorithm, 147
  tableau algorithm, 88
ambiguity, 33

**B**

bridging, 34

**C**

consistency, 7, 123
  conceptual consistency, 7, 124
  logical consistency, 7, 134

**D**

Description Logic, 10, 30, 64, 86, 135
discourse relations, 21, 34
Discourse Representation Structures, 76
distributional hypothesis, 4, 25, 44

**E**

entailment rules, 54
evaluation, 11, 17, 177

**I**

implicit predicates, 35, 185
inference machine, 2

**K**

knowledge
  dictionary knowledge, 21
  distributional knowledge, 118
  encyclopedic knowledge, 21
  lexical-semantic knowledge, 4, 40, 96
  ontological knowledge, 5, 56, 116
  world knowledge, 2, 30
knowledge base, 2, 93
knowledge representation frameworks, 27
  frames, 28
  procedural semantics, 27
  semantic networks, 28
knowledge sources, 40
  automatically generated databases
    lexical-semantic databases, 53
    ontologies learned from text, 65
    Proposition Store, 114
  electronic dictionaries, 4, 23, 46
    FrameNet, 50, 101, 126
    Propositional Bank, 49
    VerbNet, 49
    WordNet, 46, 96
  ontologies, 5, 116, 134, 166, 207
    DOLCE, 61
    domain ontologies, 64
    foundational ontologies, 59
    OpenCyc, 63
    SUMO, 62

**L**

Latent Semantic Analysis, 118
lexical-semantic relations, 23, 40, 54
lexicon-ontology mapping, 59, 116
logical form, 2, 20, 73, 95

**M**

meaning
  lexical meaning, 21
  linguistic meaning, 19
metaphor and metonymy, 35

**N**

natural language understanding (NLU), 15

domain-specific NLU, 12, 207
inference-based NLU, 8
NLU tasks
paraphrasing noun dependencies, 12, 185
recognizing textual entailment, 11, 179
semantic role labeling, 11, 48, 182
noun dependencies, 35, 114, 185

**P**

pointwise mutual information, 25, 112

**R**

reasoner, 73
DL reasoner, 166
HermiT, 89, 171
KAON2, 89
Mini-TACITUS, 10, 155, 195
Nutcracker, 10, 81, 187
reasoning, 2, 8, 73
abduction, 10, 81
weighted abduction, 82, 155, 195
deduction, 9, 77, 187

**S**

selectional preferences, 22, 54
semantic parser, 8, 74
semantic roles, 22, 48
semantic similarity, 4, 25, 118
semantic theories, 19
cognitive semantics, 22
distributional semantics, 25
formal semantics, 2, 20
discourse semantics, 20
lexical semantics, 2, 21
decompositional semantics, 21
Generative Lexicon, 24
Prototype theory, 22
similarity space, 118, 171

**T**

text interpretation, 15

**V**

vector space models, 25, 118

**W**

word sense, 43, 97
word sense disambiguation, 97