PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

A General Framework for Exploiting Background Knowledge in Natural Language Processing

Kateryna Tymoshenko

Advisor:

Claudio Giuliano

Fondazione Bruno Kessler, Human Language Technology Research Unit.

December 2012

Abstract

The two key aspects of natural language processing (NLP) applications based on machine learning techniques are the learning algorithm and the feature representation of the documents, entities, or words that have to be manipulated. Until now, the majority of the approaches exploited syntactic features, while semantic feature extraction suffered from low coverage of the available knowledge resources and the difficulty to match text and ontology elements. Nowadays, the Semantic Web made available a large amount of logically encoded world knowledge called Linked Open Data (LOD). However, extending state-of-the-art natural language applications to use LOD resources is not a trivial task due to a number of reasons, including natural language ambiguity and heterogeneity and ambiguity of the schemes adopted by different LOD resources.

In this thesis we define a general framework for supporting NLP with semantic features extracted from LOD. The main idea behind the framework is to (i) map terms in text to the unique resource identifiers (URIs) of LOD concepts through Wikipedia mediation; (ii) use the URIs to obtain background knowledge from LOD; (iii) integrate the obtained knowledge as semantic features into machine learning algorithms. We evaluate the framework by means of case studies on coreference resolution and relation extraction. Additionally, we propose an approach for increasing accuracy of the mapping step based on the "one sense per discourse" hypothesis. Finally, we present an open-source Java tool for extracting LOD knowledge through SPARQL endpoints and converting it to NLP features.

Keywords

Natural Language Processing, Information Extraction, Relation Extraction, Linked Open Data, Background Knowledge

Acknowledgments

I would like to thank all the people who made this thesis possible.

First of all, I thank my advisor, Claudio Giuliano, for his guidance, advices, availability for discussions, enthusiasm, support and encouragement. I am extremely grateful to him for helping me to learn a lot in this years and to grow professionally.

I thank my colleagues from the Human Language Technologies unit, FBK, who create an inspiring and motivating working environment. Another thanks goes to Volha Bryl, DKM unit, FBK, for the fruitful collaboration and advices.

I would like to thank Swapna Somasundaran, my internship supervisor at Siemens Corporate Research, U.S., for her guidance, valuable advices, friendliness and for our joint work. I also thank my other Siemens colleagues and the very friendly and helpful Siemens intern community.

I am grateful to the colleagues from the Web of Data unit, FBK, for their availability to help and to answer the questions.

I would like to thank Diana Maynard, Roberto Navigli and Marco Baroni for agreeing to be my thesis defense committee members, an interesting discussion after the defense and their valuable feedback.

I thank all friends whom I met during these years. I cannot list all the names here due to the space limitations. I am grateful to all my flatmates, both in Italy and U. S., for creating a warm home atmosphere.

I would like to thank Stanislav for his love and support during all these years and for constantly believing in me.

Finally and most importantly, I want to thank my parents, Marina and Valeriy Tymoshenko, for all they did for me, for their love, for always being for me, for encouraging to grow and to progress.

Contents

1 Introduction **1**

1.1 The context 1

1.2 The problem 3

1.3 The solution 4

1.4 Contributions 4

1.4.1 Coreference resolution 5

1.4.2 Semantic relation extraction between nominals. . . 6

1.4.3 Relation mining in the biomedical domain 7

1.4.4 Improving text-to-Wikipedia mapping by expanding internal link annotations in Wikipedia pages 8

1.5 Structure of the thesis 8

2 Introduction to Linked Open Data **11**

2.1 Introduction 11

2.2 Origins of the LOD 12

2.3 LOD principles 16

2.4 Consuming Linked Open Data 19

2.4.1 Resource Description Framework 19

2.4.2 Accessing LOD data 20

2.4.3 SPARQL query language 20

2.4.4 Processing RDF data 22

2.5 Overview of the LOD content 22

2.5.1 Frequently used data models 23
2.5.2 DBpedia 25
2.5.3 YAGO 27
2.5.4 Freebase 28
2.5.5 WordNet 29
2.5.6 Cyc 30

3 The Framework Implementation 31
3.1 Introduction 31
3.2 Overall picture 32
3.3 The Wiki Machine 34
3.3.1 Training set 35
3.3.2 Learning algorithm 35
3.3.3 Implementation details 37
3.3.4 Evaluation 37
3.3.5 Related work 39
3.4 LOD-based semantic feature extraction schema and implementation details 42
3.4.1 Extracting LOD data relevant for feature extraction 42
3.4.2 Extracting features 43

4 Coreference resolution 47
4.1 Introduction 47
4.2 Related work 49
4.2.1 Machine learning approaches to coreference resolution 51
4.2.2 Semantic features employed for co-reference resolution 52
4.3 Background knowledge (BK) acquisition 58
4.3.1 Feature extraction 59
4.3.2 Feature selection 60
4.4 Experiments 61

4.4.1 Baseline model definition 61
4.4.2 Injecting background knowledge into coreference model 66
4.5 Conclusion and future work 70

5 Semantic relation extraction between pairs of nominals 71
5.1 Introduction 72
5.2 Related work 74
5.3 Kernel methods for Relation Extraction 77
5.3.1 Shallow syntactic kernels 77
5.3.2 Semantic kernels 79
5.3.3 Semantic kernel instantiation 80
5.4 Experiments 83
5.4.1 Experimental setup 83
5.4.2 BK enrichment evaluation and discussion 84
5.4.3 SRE experiments and discussion 88
5.5 Conclusion 91

6 Biomedical entity relation mining 93
6.1 Introduction 94
6.2 Entity-level semantics 96
6.3 Semantic features 98
6.3.1 Entity-specific features 98
6.3.2 Entity pair linkage features 100
6.4 Experiments 100
6.4.1 Data 101
6.4.2 Baseline 102
6.4.3 Entity-level semantics (ELS) systems 105
6.4.4 Ensemble of entity-level semantics classifiers 107
6.5 Discussion 108
6.6 Related work 110

6.7 Conclusion 113

7 Improving linking to Wikipedia 115

7.1 Introduction 115

7.2 Adapting the One Sense Per Discourse hypothesis to Wikipedia 117

7.3 Experiments 120

7.3.1 Test set 121

7.3.2 One sense per article procedure evaluation 123

7.3.3 One sense per category procedure evaluation 124

7.4 Discussion 125

7.5 Conclusion 126

8 Conclusions 129

Bibliography 135

List of Tables

2.1 RDF statements examples 19

2.2 Linked Data by domain 23

3.1 Comparative evaluation of the two disambiguation methods on ACE05-WIKI (micro-average). Symbol † indicates significant differences relative to the corresponding mention type ($p < 0.01$). Significance tests are computed using approximate randomization procedure. 39

4.1 Feature examples 60

4.2 Feature examples 60

4.3 Selected features 67

4.4 MUC scores for GPE and PER NE types, *ACE-SUBSET-2* document set 69

4.5 MUC scores for GPE and PER NE types, *ACE-SUBSET-3* document set 69

5.1 Notation 80

5.2 Coverage 85

5.3 TWM performance (in %) 85

5.4 Results of manual analysis of mappings produced by the framework 86

5.5 Overall performance on the test set, macro-average over all relation excluding "other". † indicates significant differences ($p < 0.05$). Significance tests are computed using approximate randomization procedure. 89
5.6 Performance in 10-fold cross-validation on the training set, macro-average over all relations excluding "other" 90
5.7 Per-relation performance on the test set in terms of F_1 measure. Value in parentheses in the *SL+WordNetAll* column corresponds to the relative improvement as compared to *SL*. Value in parentheses in the *SL+WordNetAll+OpenCycAll* column corresponds to the relative improvement as compared to *SL+WordNetAll* 90

6.1 Relations of interest from NDF-RT 97
6.2 Baseline system performance. 103
6.3 Performance of best feature sets per relation on test instances covered by a specific feature set. R and F_1 for the same feature set on the full data set are reported in parentheses (P remains the same under both conditions). . . . 104
6.4 Performance of ensemble and STCUI baseline systems. Overall is obtained by macro-averaging over results for individual relations. 109
6.5 Distance Supervision - Using STCUI 111

List of Figures

2.1 The Semantic Web Stack 15
2.2 Part of DBpedia RDF graph describing Douglas Adams . . 18
2.3 Part of the LOD cloud 24

3.1 Overall schema of extracting LOD knowledge and converting it to NLP features. 32

7.1 The ratio between the number of acquired and existing labeled examples. 124
7.2 The performance comparison between the different disambiguation models: (a) in page label propagation procedure, (b) in-category label propagation procedure. Ace Y corresponds to the accuracy; ace X corresponds to the fraction of labeled training data used for the propagation propagation procedure. 127

Chapter 1

Introduction

1.1 The context

Natural language processing (NLP) field is concerned with developing automatic systems capable of "understanding" natural human language, e.g. answering questions or converting human language into structured commands.

State-of-the-art approaches to NLP tasks are based on machine learning. The two key aspects of NLP applications based on machine learning techniques are the learning algorithm and the feature representation of the documents, entities, or words that have to be manipulated. Both aspects are important. Reviewing the relevant literature of the last years, one realizes that, on the one hand, the difference between the results obtained by different learning algorithms (e.g., support vector machines vs. decision trees) is statistically significant when they are fed with the same information. On the other hand, feature extraction and representation methods also play a crucial role for the accuracy of the system. For example, in relation extraction approaches that exploit deep syntactic parsing outperform the ones that use only shallow syntactic analysis.

The majority of NLP features encode properties and relations of *words* in text in consideration, e.g. bag-of-words, syntactic information such as

part-of-speech tags, syntactic constituency information or grammar dependency relations. However, in many works it has been shown that performance of NLP algorithms considerably improves when one employs features encoding implicit or **background knowledge** about *concepts or individuals* mentioned in a text [Ponzetto and Strube, 2006, Chan and Roth, 2010, Soon et al., 2001, Zhou et al., 2005]. For example, consider the sentence:

> *"Towel Day:* ***Douglas Adams*** *Fans Celebrate Late Hitchhikers Guide To The Galaxy* ***Author.****"*[1]

Here knowledge that "Douglas Adams" is a noun phrase, "fans" is a possession modifier of "Adams" is knowledge about words and text, while knowledge that "Douglas Adams" is a writer and writer is an author is background knowledge about an individual referred to as "Douglas Adams" and a concept referred to as "writer".

After reading the sentence, a human not familiar with the subject would not understand that the mentions *[Douglas Adams]* and the *["Hitchhikers Guide to The Galaxy" author]* refer to the same entity. Similarly, a state-of-the-art coreference resolution system[2] might not detect that the mentions are coreferent. Intuitively, background knowledge that Douglas Adams is an author (or even more specific knowledge that he is the author of the book called "Hitchhikers Guide to The Galaxy") can help a human reader and even more so an automatic system to resolve the coreference. While humans can query the World Wide Web for the unfamiliar names, automatic systems need structured data. Such data can be obtained from external structured sources such as knowledge bases.

[1] From `http://www.inquisitr.com/242961/towel-day-douglas-adams-fans-celebrate-late-hitchhikers-guide-to-the-galaxy-author/\#G7m0bVp0ftR1w47W.99`

[2] We processed the sentence with the online version of the Stanford CoreNLP toolkit: `http://nlp.stanford.edu:8080/corenlp/`, and it did not detect any coreference.

1.2 The problem

So far,[3] background knowledge extracted from knowledge bases has been restricted to WordNet [Fellbaum et al., 1998], ad-hoc gazetteers and, more recently, Wikipedia. Problems typically encountered were the low coverage of the available knowledge resources and the difficulty to match text and ontology elements. Recently, Wikipedia[4] became a partial solution for the problem of coverage [Ponzetto and Strube, 2006], however, it lacks formal ontological structure.

Nowadays, the Semantic Web made available a large amount of logically encoded information (e.g., ontologies, RDF(S)-encoded knowledge bases, etc.) called Linked Open Data (LOD) which constitute a valuable source of semantic knowledge. However, extending the state-of-the-art NLP applications to use these resources is not a trivial task due to the following reasons:

1. The *ambiguity* of natural human language. Semantic Web knowledge is concept-level, hence different meanings of an ambiguous word may refer to different concepts.

2. The *heterogeneity* and the *ambiguity* of the schemes adopted by the different resources of the Semantic Web. This means, e.g., that the same relation can be encoded by different unique resource identifiers (URIs).

These issues define our research directions.

[3] "so far" refers to the year of thesis proposal submission, that is 2010

[4] `http://www.wikipedia.org/`

1.3 The solution

In this thesis we define a general framework for supporting natural language processing with background knowledge available in the LOD and propose practical solutions for the aforementioned problems. The framework can be described as follows.

First, we map terms in text to the Semantic Web concepts' URIs through Wikipedia mediation. We benefit from the fact that most of the resources available in the LOD are aligned with Wikipedia on concept level, so it can be used as a *semantic mediator*. Therefore, we propose to link text to Wikipedia articles and then to exploit the linking between Wikipedia and the other resources to access the knowledge encoded in them. Wikipedia represents a practical choice as it is playing a central role in the development of the Semantic Web. The large and growing number of resources linked to it makes Wikipedia one of the central interlinking hubs of the Linked Open Data.

Second, we query the LOD using the URIs to obtain the background knowledge expressed in the RDF/OWL formalism. We select relevant knowledge manually or apply feature selection techniques to retrieve knowledge relevant for a specific task.

Finally, we integrate the obtained knowledge as features into machine learning algorithms.

1.4 Contributions

The contribution of this thesis is the idea of the general framework for using semantic features extracted from background knowledge from LOD resources in NLP tasks and recommendations for its implementation. We have conducted case studies in the tasks of coreference resolution between

nominal and named entity mentions (see Section 1.4.1); semantic relation extraction between pairs of nominals (see Section 1.4.2) and relation mining between biomedical entities such as drugs and diseases (see Section 1.4.3).

Finally, we have developed a Java tool for extracting LOD knowledge through SPARQL endpoints, storing the knowledge locally, and extracting semantic features from the local knowledge repository. We plan to release this tool as an open-source.

Subsections below list our contributions in specific case studies.

1.4.1 Coreference resolution

We combine semantic information available in LOD with statistical methods for the coreference resolution task, using Wikipedia as a semantic mediator between text and LOD. LOD sources are represented by YAGO, Freebase and DBpedia, while the machine learning method employed is MLN, that is Markov Logic Networks. The results show that background knowledge helps to increase the overall MUC F_1 measure due to the increase in recall. This work has led to the following publications:

- Volha Bryl, Claudio Giuliano, Luciano Serafini, **Kateryna Tymoshenko**. "Using Background Knowledge to Support Coreference Resolution." In *Proceedings of 19th European Conference on Artificial Intelligence (ECAI 2010)*, pp. 759–764, Lisbon, Portugal, 2010.

- Volha Bryl, Claudio Giuliano, Luciano Serafini, **Kateryna Tymoshenko**. "Supporting Natural Language Processing with Background Knowledge: Coreference Resolution Case." In *Proceedings of the 9th International Semantic Web Conference (ISWC2010)*, Shanghai, China, 2010.

- Luisa Bentivogli, Claudio Giuliano, Pamela Forner, Alessandro Marchetti, Emanuele Pianta, **Kateryna Tymoshenko**. "Extending English

ACE 2005 Corpus Annotation with Ground-truth Links to Wikipedia." In *Proceedings of the 2nd Workshop on The Peoples Web Meets NLP: Collaboratively Constructed Semantic Resources, Coling 2010.*, pp. 19–27, Beijing, China, 2010.

This work has contributed to the collaboration resulting in the following publications (which are not included into the thesis):

- Olga Uryupina, Massimo Poesio, Claudio Giuliano, **Kateryna Tymoshenko**. "Disambiguation and Filtering Methods in Using Web Knowledge for Coreference Resolution". In *Proceedings of 24th International FLAIRS Conference*, pp. 317–322, Florida, USA, 2011.
- Olga Uryupina, Massimo Poesio, Claudio Giuliano, **Kateryna Tymoshenko**, "Disambiguation and Filtering Methods in Using Web Knowledge for Coreference Resolution", in *Chutima Boonthum-Denecke , Philip M. McCarthy, Travis Lamkin (eds.), Cross-Disciplinary Advances in Applied Natural Language Processing: Issues and Approaches,* Hershey, IGI Global , 2011, pp. 185 – 201

1.4.2 Semantic relation extraction between nominals.

We have shown how semantic relation extraction between nominals can be improved by combining background knowledge with shallow syntactic processing. Background knowledge is obtained from WordNet, OpenCyc and YAGO. We use kernels measuring similarity of pairs of nominals within a context in terms of shallow syntactic features and define new kernels operating upon semantic properties of the nominals. The approach is shown to be state-of-the-art ranking 2^{nd} in the Task 8, "Semantic relation between pairs of common nominals", during the SemEval 2010 evaluation campaign.

This work has led to the following publication:

- **Kateryna Tymoshenko** and Claudio Giuliano. "FBK-IRST: Semantic Relation Extraction Using Cyc." In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010)*, pp. 214–217, Uppsala, Sweden, 2010.

1.4.3 Relation mining in the biomedical domain

In this work[5] we have explored the use of semantic information from background knowledge sources for the task of relation mining between medical entities such as diseases, drugs, and their functional effects/actions. When conducting this research we have discovered that the biomedical resources currently available on LOD have limited coverage for our medical entities of interest due to the proprietary nature of the data in the domain.

Therefore, in this research direction we have deviated from the first two steps of the framework, and employed alternative ways of extracting knowledge. We extract features from Wikipedia and specialized biomedical resources, including UMLS Semantic Network, MEDCIN, MeSH and SNOMED CT. Given that the resources might have different coverage, we propose a two-step approach. First, we learn multiple classifiers combining features from different resources, and correspondingly having different amount of semantic knowledge/coverage balance. Then we combine the predictions of the individual classifiers by means of an ensemble classifier. We show than in contrast to the general domain, semantic features can be highly discriminative, even in absence of syntactic evidences.

This work has led to the following publication:

- **Kateryna Tymoshenko**, Swapna Somasundaran, Vinodkumar Prabhakaran, Vinay Shet. "Relation Mining in the Biomedical Domain

[5]The author conducted this study while she was at Siemens Corporate Research, Princeton, New Jersey, U.S., for an internship under supervision of Dr. Swapna Somasundaran.

using Entity-level Semantics." *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, Montpellier, France, 2012.

1.4.4 Improving text-to-Wikipedia mapping by expanding internal link annotations in Wikipedia pages

In our framework Wikipedia is a mediator between text and background knowledge, therefore the quality of linking text to Wikipedia articles constitutes an important factor in its overall performance. The last part of the thesis is concerned with improving the accuracy of this linking.

We annotate text with links to Wikipedia using a supervised Wikipedia-based word sense disambiguation system. It is trained on the labeled data automatically extracted from the Wikipedia internal link annotations. However, the distribution of the data is highly skewed, e.g., rare senses often have a lot of examples, while more frequent ones are sometimes absent.

We propose an approach based on applying the one sense per discourse hypothesis to Wikipedia pages and categories in order to automatically expand Wikipedia annotations. Experiments show that the hypothesis is generally correct within Wikipedia allowing us to improve disambiguation accuracy on a benchmark data set.

1.5 Structure of the thesis

The remainder of the thesis is structured as follows:

Chapter 2 introduces the idea and the principles of Linked Open Data (LOD), major LOD resources relevant for the further chapters, major techniques of handling RDF data from LOD.

Chapter 3 describes the details of our instantiation of the framework,

namely the approach that we employ to map plain text to Wikipedia and technical aspects of extracting LOD knowledge by the mediation of the Wikipedia links.

Chapter 4 describes the application of the framework to the task of coreference resolution.

Chapter 5 describes application of the framework to the task of semantic relation extraction between nominals.

Chapter 6 describes usage of background knowledge in the task of biomedical relation mining.

Chapter 7 describes our methodology for increasing the amount of training data for a Wikipedia-based word sense disambiguation system.

Chapter 8 draws the conclusions.

Given that this thesis is cross-task and cross-domain, we do not have a single state-of-the-art chapter. Instead, in each chapter we propose an overview of related work for a corresponding task and domain.

Chapter 2

Introduction to Linked Open Data

This chapter aims to give introduction to Linked Open Data (LOD).

2.1 Introduction

Linked Data is a paradigm under which structured data are published on the Web by different data providers who use standard formats and vocabularies. Similarly to the HTML web-documents, these data can be dereferenced by means of HTTP protocol, and datasets provided by the different data contributors are interlinked. Thus Linked Data can be viewed as a global data space, a web of data, organized similarly to the web of documents, but in the contrast to the latter destined to be used by the automatic agents. Freely available Linked Data datasets constitute the Linked Open Data (LOD).

In this chapter we aim to provide the general introduction to LOD, including its origins (Section 2.2), main principles (Section 2.3), terminology and mechanisms (Section 2.4), core datasets and vocabularies (Section 2.5).

2.2 Origins of the LOD

The origins of LOD trace back to 2001. At that time World Wide Web (WWW) was a web of text documents interconnected by means of untyped hyperlinks. It did contain structured data as well, but they were published in multiple different formats, e.g. CSV, XML documents or HTML tables. Some services, like Amazon, provided APIs that sent structured data encoded in a micro-format in a response to a structured query. However, formats of documents or API queries and responses varied from provider to provider. The other problem was different semantics of the structured sources, e.g. two fields named "Address" in two different databases do not necessarily contain the same data. For example, one database could contain geographic coordinates, while another listed human readable post addresses. Format heterogeneity and absence of semantics made the task of automatic accessing and processing structured data challenging as each data source had to be processed separately, with its data format and semantics taken into account.

Let us describe what this means in our specific use case of enriching plain-text documents with background semantic knowledge. Consider the following snippet:

> "Towel Day: [**Douglas Adams**]$_{EM1}$ Fans Celebrate Late [**'Hitchhikers Guide To The Galaxy'**]$_{EM3}$ [**Author**]$_{EM2}$"[1].

Intuitively, knowledge that *Douglas Adams* is a writer (along with syntactic and contextual information) might be helpful to establish that *EM1* and *EM2* refer to the same entity. Knowing that *EM3* is a book would be a helpful feature to extract the fact `(DouglasAdams,authorOf,Hitchhikers-`

[1] From `http://www.inquisitr.com/242961/towel-day-douglas-adams-fans-celebrate-late-hitchhikers-guide-to-the-galaxy-author/\#G7m0bVp0ftR1w47W.99`

GuideToTheGalaxy). Back in 2001, in order to enrich text with such knowledge one would have to use multiple tools and resources. Gazetteer of names (GZN) or a NER tool (NERT) would provide intuition that Douglas Adams is a person; navigating the WordNet (WN) lexical database would help to understand that "author" is a hyponym of a "person". Alternatively API of a bookstore (API) could provide information that Douglas Adams is an author. NERT or an API would give intuition that the *EM3* as an artifact or a book.

A set of problems may arise when obtaining the information from the above-mentioned resources. First, GZN, NERT, WN, API would require different software to extract this kind of knowledge because of their different data formats and access mechanisms. Second, we would need to study the semantics of each source in detail in order to understand which data from it are useful, and what do specific fields (or labels in case of NERT) mean.

Finally, in case of GZN, WN and API we would have to solve the problem of ambiguity and lack of coverage. For example, there are other famous people called Douglas Adams, including an English professor[2] and a music journalist.[3] Moreover, there is also a Douglas township in Adams County, in Iowa.[4] Gazetteer of geographical names might contain the latter, and thus we are likely to wrongly assume that *EM1* is a location. WN does not know any Douglas Adams, but it has four possible meanings for the surname *Adams*. Finally, even if we manage to find a reference to Douglas Adams in a bookstore, we are not guaranteed that he is not a namesake of the Adams mentioned in the text. We could try to solve the ambiguity problem by building word sense disambiguation (WSD) systems. However, since all the above-mentioned background knowledge sources have different

[2] http://en.wikipedia.org/wiki/Douglas_Q._Adams

[3] http://en.wikipedia.org/wiki/Doug_Adams_(music_journalist)

[4] http://en.wikipedia.org/wiki/Douglas_Township,_Adams_County,_Iowa

sense inventories and different structures, we would need to create a WSD system for each knowledge source separately.

In 2001 Tim Berners-Lee formalized the limitations of the WWW of those days in the perspective of automatic processing: it was intended mostly for human use or for the use by very specialized agents [Berners-Lee et al., 2001]. He proposed an idea of extending the WWW with machine-readable data, published in a decentralized fashion by independent data providers, and inference mechanisms on the top of these data. Berners-Lee et al. [2001] provided a use-case of an automatic agent, scheduling a visit to a doctor for the mother of a hypothetic user. In order to deliver a solution the agent would need to retrieve, understand and analyze the schedules of the doctor and the user, to take into account the distance from the user's home to the doctor's office. This can be achieved by using standard approaches to describing knowledge (e.g. Resource Description Framework (RDF) (see Section 2.4.1)), standard data formats, unambiguous identifiers for things and ontologies describing the world in a standardized language.

By 2006, the e-science communities accepted the idea and created a number of new ontologies [Shadbolt et al., 2006]. At the same time a set of organizations, including World Wide Web Consortium (W3C), devised the Web Ontology Language (OWL) to describe the complex models, RDFS (Resource Description Framework Schema) to describe simpler things, and various reasoners and rule exchange formats to support the inference. Triple stores permitted to store and index RDF data, and SPARQL query language was designed to extract information on demand. Figure 2.1[5] demonstrates how these components are integrated into the global view of the Semantic Web (SW) architecture. However, despite the technological advances of SW, Shadbolt et al. [2006] stressed that the semantic web technologies lacked "real viral uptake" and the "data exposure

[5]picture is taken from `http://www.w3c.it/talks/2005/openCulture/slide7-0.html`

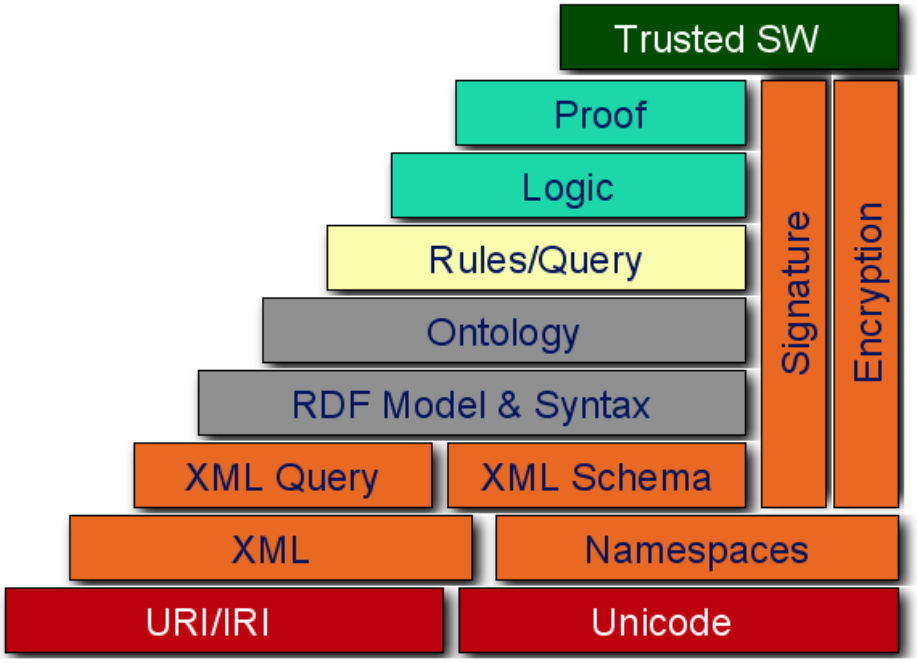

Figure 2.1: The Semantic Web Stack

revolution had not yet happened."

In the same year, the online document by Berners-Lee [2006] noted that SW is also about linking data to each other. Berners-Lee [2006] published a set of principles to be observed by the data contributors to SW. Data published under observance of these principles constitute the so-called Linked Data (LD). Linked data published under an open license are called Linked Open Data (LOD). In his presentations he stated that Linked Data is "the Semantic Web done right".[6]

According to the official document published by W3C "*Linked Data lies at the heart of what Semantic Web is all about: large scale integration of, and reasoning on, data on the Web.*"[7] Nowadays, SW is called also the Web of Data, as the name "Semantic Web" is often considered difficult to

[6] http://www.w3.org/2008/Talks/0617-lod-tbl/#(3)
[7] http://www.w3.org/standards/semanticweb/data

understand and misleading.

In the following subsections we will consider the mechanisms behind the LD in more detail and overview some of the LOD resources.

2.3 LOD principles

Berners-Lee [2006] introduced the concept of the Linked Data (LD). It is the web of structured descriptions of "things" interconnected by links. He listed the main principles to be observed by the LD publishers.[8]

1. Use URIs as names for things.

2. Use HTTP URIs so that people can look up those names.

3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).

4. Include links to other URIs. so that they can discover more things.

The first principle means that each data publisher must give a unique name, called a Unique Resource Identifier (URI), to each "thing" from his/her dataset. The second principle suggests that given a URI a user or a machine must be enabled to retrieve the corresponding resource by means of the HTTP protocol. Naturally, if the resource is a physical thing it cannot be transmitted by HTTP, but it is possible to transmit the description of this thing. Heath and Bizer [2011] describe the technical details in their book.

Let us consider the principles in detail using our example from the previous section. For convenience we repeat it here:

[8]The list below is a verbatim quote from `http://www.w3.org/DesignIssues/LinkedData.html`

> "Towel Day: [**Douglas Adams**]$_{EM1}$ Fans Celebrate Late [**Hitchhikers Guide To The Galaxy**]$_{EM3}$ [**Author**]$_{EM2}$"[9].

URIs for EM1, EM2, EM3 taken from DBpedia (one of the core resources in the LOD, see Section 2.5.2) are `http://dbpedia.org/resource/Douglas_Adams`, `http://dbpedia.org/resource/The_Hitchhiker's_Guide_to_the_Galaxy` and `http://dbpedia.org/resource/Author`. Note that the writer's namesakes have different URIs, e.g. `http://dbpedia.org/resource/Douglas_Q._Adams` for the English professor and `http://dbpedia.org/resource/Doug_Adams_(music_journalist)` for the music journalist. If one pastes these links into the address line of a browser he or she will retrieve the descriptions of the corresponding "things".

Principle 3 means that if one dereferences the URI he or she should be provided with information considered useful by the data publisher. The information will be encoded in the Resource Description Framework (RDF) formalism (see Section 2.4.1). For example, when dereferencing `http://dbpedia.org/resource/Douglas_Adams` we obtain information about the type of the entity, genres he worked in, list of his books, his full given name and other kinds of information. Figure 2.2 shows a subgraph of useful information about the writer, available in DBpedia. In the context of our example this information is the background knowledge we require to extract NLP features.

Finally, the last principle means that the data providers should link their URIs to URIs in the other datasets. For instance, in the Figure 2.2 the `rdf:type` link connects the Adams' URI to the `foaf:Person`. The latter is a URI corresponding to the abstract notion of a person in the *The Friend of a Friend* (FOAF) ontology, that is independent from DBpedia. Therefore, among other things, linking allows datasets and applications

[9]From `http://www.inquisitr.com/242961/towel-day-douglas-adams-fans-celebrate-late-hitchhikers-guide-to-the-galaxy-author/\#G7m0bVp0ftR1w47W.99`

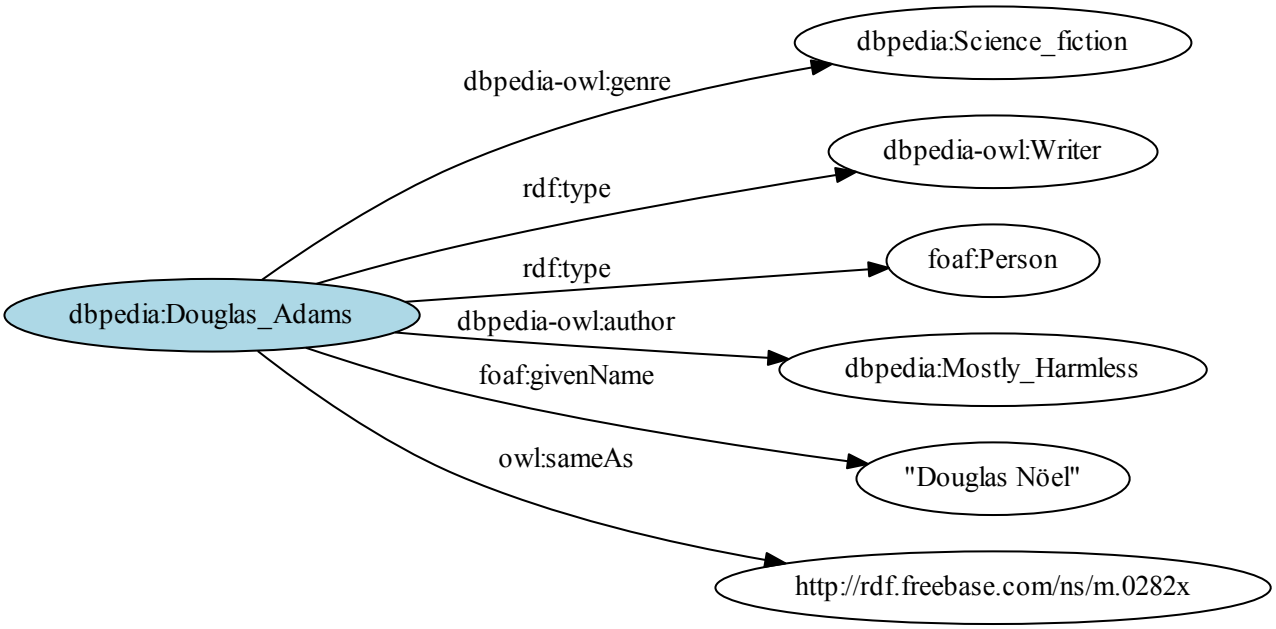

Figure 2.2: Part of DBpedia RDF graph describing Douglas Adams

to use common vocabularies for the high-level concepts, e.g. *person* or *location*.

One of the most important links is the `owl:sameAs` link. Typically, different LOD datasets have different URIs (and correspondingly different descriptions) for the same thing. For example, Freebase (large-scale database collaboratively constructed by the users, see Section 2.5.4) URI for the writer Douglas Adams is `http://rdf.freebase.com/ns/m.0282x`. In Figure 2.2 `owl:sameAs` link connects the DBpedia URI to the Freebase URI. We can dereference the Freebase URI and retrieve the alternative description of the writer from Freebase. `owl:sameAs` links are the LOD "glue". Using them we can navigate between datasets published by different providers and obtain different facts about the same entity.

Subject	Predicate	Object
`dbpedia:Douglas_Adams`	`rdf:type`	`dbpedia-owl:Writer`
`dbpedia:Douglas_Adams`	`foaf:givenName`	*"Douglas Noel"*

Table 2.1: RDF statements examples

2.4 Consuming Linked Open Data

In this section we will briefly describe the RDF data format and the conventional ways to get access to LOD.

2.4.1 Resource Description Framework

Resource Description Framework (RDF)[10] is a model for describing data. In RDF data are described as a set of statements consisting of *subject*, *predicate* and *object*. A predicate describes a directed relationship between a subject and an object. All three may be identifiers, i.e. URIs, of the other resources. Objects may also be typed literals, e.g. integers, strings, and plain (untyped) literals. Subjects and objects may be blank literals typically needed for technical purposes, e.g. for encoding complex attributes. We provide some examples of RDF statements in Table 2.1. Note that predicates are resources, and may be dereferenced in order to obtain their description. Alternatively, one can regard an RDF description as a directed graph where predicates are directed labeled edges which connect subject and object nodes. Figure 2.2 shows an example of such graph.

Currently, there exists a number of standard RDF serializations understood by the main Semantic Web processing engines. They include RDF/XML[11] serialization, Turtle,[12] N-Triples.[13]

[10] `http://www.w3.org/RDF/`
[11] `http://www.w3.org/TR/rdf-syntax-grammar/`
[12] `http://www.w3.org/TeamSubmission/turtle/`
[13] `http://www.w3.org/TR/rdf-testcases/#ntriples`

2.4.2 Accessing LOD data

In order to use the LOD datasets we can download them locally. However, in many cases this is excessive, because the datasets might be large, and we might need only a subset of knowledge, e.g. information about few "things". [Heath and Bizer, 2011] in Section 6.3 of their book summarize the major patterns used to obtain a subset of relevant data from LOD as follows:

1. **Crawling.** Obtain data by consequentially dereferencing URIs (e.g. LDSpider [Isele et al., 2010]). This is similar to browsing WWW by following hyperlinks. In case of LOD this means dereferencing a URI, examining the obtained RDF statements, and then dereferencing other components of these statements if needed.

2. **Dereferencing on-fly.** Run complex queries on LOD, by dereferencing multiple URIs on-fly and on-demand, e.g. [Hartig et al., 2009].

3. **Query federation.** Sending queries to multiple public SPARQL (see Section 2.4.3) endpoints.

2.4.3 SPARQL query language

SPARQL (SPARQL Protocol and RDF Query Language)[14] is a query language designed for querying RDF graphs.

A SPARQL query is a graph pattern. The simplest example of a graph pattern is the *basic graph pattern* (BGP). BGP can be regarded as a set of RDF statements or an RDF graph in which some elements are uninitialized variables.

The following is an example of a query against the DBpedia graph:

[14] `http://www.w3.org/TR/rdf-sparql-query/`

```
PREFIX rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT distinct ?type
FROM <http://dbpedia.org>
where {
      <http://dbpedia.org/resource/Douglas_Adams> rdf:type ?type
}
```

Here keyword "PREFIX" specifies a shorthand for the frequently used parts of URIs designed to make queries more readable. "FROM" is an optional keyword which specifies the name of the RDF graph in a repository on which the query should be resolved. *?type* is the variable that should be bound to concrete values in a resolved query.

Resolving a SPARQL query against an RDF graph, means finding subgraphs of this graph that match the graph pattern expressed by the query. For example, one of the DBpedia (see Section 2.5.2) subgraphs matching the query above would be the triple

```
(dbpedia:Douglas_Adams, rdf:type, dbpedia-owl:Writer).
```

Here variable `?type` is bound to the URI `dbpedia-owl:Writer`.

Remote RDF graphs can be queried through SPARQL endpoints. A SPARQL endpoint *"is a conformant SPARQL protocol service as defined in the SPROT*[15] *specification."*[16] Many large-scale knowledge stores have publicly available SPARQL endpoints. For example, the SPARQL endpoint of one of the core LOD datasets, DBpedia, can be accessed at `http://dbpedia.org/sparql/`. Lately, OpenLink Software[17] made available the *OpenLink Software LOD Cache*[18] mirroring a number of LOD resources, e.g. YAGO, OpenCyc and WordNet (see Section 2.5) .

[15] SPROT stands for SPARQL Protocol for RDF (`http://www.w3.org/TR/rdf-sparql-protocol/`)
[16] `http://semanticweb.org/wiki/SPARQL_endpoint`
[17] `http://www.openlinksw.com/`
[18] http://lod.openlinksw.com/sparql

2.4.4 Processing RDF data

A number of tools allows to operate the RDF data. Two of the most popular tools are the Java-based Apache Jena Framework[19] and Sesame Framework.[20] They provide utilities for creating and modifying various data models, including RDFS and OWL (see Section 2.5.1), performing inference on them, performing SPARQL queries on both local and remote resources.

We can store RDF data locally in several ways. First, data can be stored as RDF files and uploaded directly into the RAM to be processed. However, if the data are large-scale it is more reasonable to store them in an index called a *triple store*. Triple stores allow to store and quickly access the large-scale data. They may have their own storage mechanism implementation, e.g. Jena TDB,[21] Virtuoso,[22] AllegroGraph,[23] Sesame,[24] or use a third party storage implementation, e.g. a common relational database management system. For example, Jena SDB[25] uses SQL.

2.5 Overview of the LOD content

Currently, the amount of Linked Data grows rapidly. Contributors to the W3C Linking Open community project[26] are concerned with making their datasets available in RDF format and connecting them to the other datasets in compliance with the LOD principles. Current state of LOD is

[19] http://jena.apache.org/
[20] http://www.openrdf.org/
[21] http://jena.apache.org/documentation/tdb/index.html
[22] http://virtuoso.openlinksw.com/
[23] http://www.franz.com/agraph/allegrograph/
[24] http://www.openrdf.org/
[25] http://jena.apache.org/documentation/sdb/index.html
[26] http://linkeddata.org/, http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData#Project_Description

Domain	Datasets	Triples	%	(Out-)Links	%
Media	25	1,841,852,061	5.82%	50,440,705	10.01%
Geographic	31	6,145,532,484	19.43%	35,812,328	7.11%
Government	49	13,315,009,400	42.09%	19,343,519	3.84%
Publications	87	2,950,720,693	9.33%	139,925,218	27.76%
Cross-domain	41	4,184,635,715	13.23%	63,183,065	12.54%
Life sciences	41	3,036,336,004	9.60%	191,844,090	38.06%
User-generated content	20	134,127,413	0.42%	3,449,143	0.68%
	295	31,634,213,770		503,998,829	

Table 2.2: Linked Data by domain

visualized in the so-called Linking Open Data cloud diagram.[27] Figure 2.3 shows a part of the diagram.[28] Here bubbles correspond to the datasets, and edges correspond to the links between datasets. Additionally, the datasets available under Linked Open Data are catalogized on *the Data Hub* website.[29]

The LOD datasets may be cross-domain or belong to a specific domain, e.g. media, geography, life sciences. Table 2.2[30] shows the state of the LOD by 19/09/2011. In the following subsections we describe the most important vocabularies employed in LOD and the largest cross-domain datasets.

2.5.1 Frequently used data models

A vocabulary defines concepts and relationships, called *terms*, to be used to organize and describe knowledge [voc, 2012]. Below we provide brief descriptions of the vocabularies (also referred to as ontologies) widely ac-

[27] `http://richard.cyganiak.de/2007/10/lod/imagemap.html`

[28] We could not include the entire cloud due to its large size. The original figure is available at `http://richard.cyganiak.de/2007/10/lod/lod-datasets_2011-09-19_colored.png`

[29] `http://thedatahub.org/group/lodcloud`

[30] Table is taken from `http://www4.wiwiss.fu-berlin.de/lodcloud/state/`

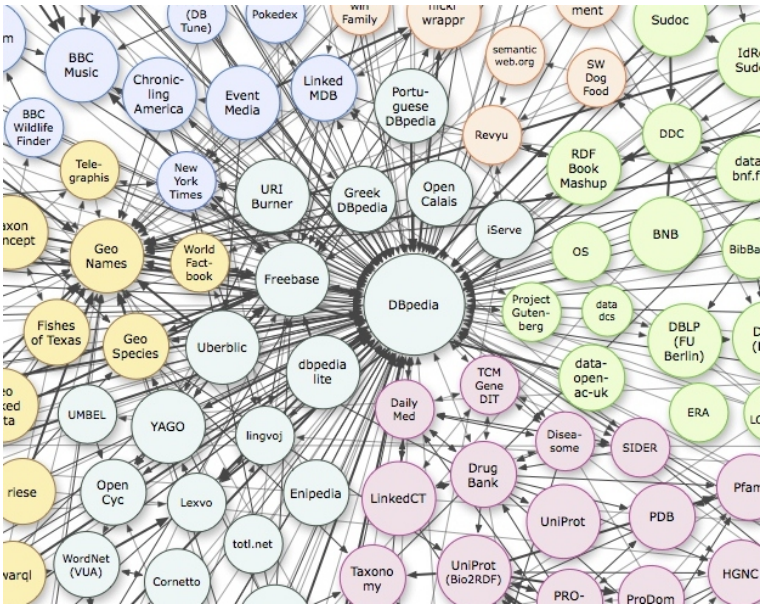

Figure 2.3: Part of the LOD cloud

cepted in the LOD community.

RDFS. Resource Description Framework Schema (RDFS)[31] is a basic shallow vocabulary. It contains basic elements for definition of other vocabularies. For example, in includes predicate `rdf:type`, used to indicate that a given instance is member of a class, and `rdfs:subClassOf`, used to describe the class taxonomy. Another widely exploited predicate is `rdfs:label` used to describe a human readable name of a resource.

OWL. Web ontology language or OWL[32] is a language for describing the data with maximum expressiveness. It is available in three versions, OWL Lite, OWL DL and OWL Full, listed in the order of their expressiveness. Inter alia, OWL provides inventory for describing classes and properties, defining new classes on tops of existing ones, expressing relations of equivalence and non-equivalence. The most important term of this vocabulary

[31] `http://www.w3.org/TR/rdf-schema/`
[32] `http://www.w3.org/TR/owl-features/`

for our purposes is `owl:sameAs` that interlinks two individuals or concepts.

SKOS. Simple Knowledge Organization System or SKOS[33] is a vocabulary for the knowledge organization systems such as nomenclatures or libraries. Its basic unit is a *concept*, `skos:Concept`. SKOS provides inventory to organize concepts into schemes, connect them by semantic relations, e.g. `skos:broader` or `skos:narrower`, record their preferred, `skos:prefLabel`, and alternative labels, `skos:altLabel`, and other utilities.

Naturally, most of the resources define their specialized vocabularies for describing more specific things, however, many of them use the vocabularies listed above.

2.5.2 DBpedia

DBpedia [Bizer et al., 2009] is a large-scale knowledge base automatically extracted from Wikipedia. Wikipedia pages are included to DBpedia as "things". A "thing" is assigned to a unique URI created by adding Wikipedia page name to the DBpedia prefix "`http://dbpedia.org/resource/`." For instance, Wikipedia page about Douglas Adams, `http://en.wikipedia.org/wiki/Douglas_Adams`, is used to create a DBpedia URI, namely `http://dbpedia.org/resource/Douglas_Adams`. Evidently, this means that given a Wikipedia page name we can easily convert it into DBpedia URI.

The core DBpedia content is created by converting Wikipedia infoboxes to RDF triples. DBpedia URI corresponding to a page with an infobox serves as a subject of a triple. Predicates and objects are obtained by means of *generic* or *mapping-based* extraction. *Generic extraction* converts names of infobox attributes into URIs of properties by adding the

[33] `http://www.w3.org/2004/02/skos/`

`http://dbpedia.org/property` to their names. Corresponding values of attributes are converted into objects. Their types (e.g., literal, typed literal or URI) are defined heuristically. The main problem here is that the Wikipedia infobox attribute names do not use the same vocabulary, and this results in multiple properties having the same meaning but different names and vice versa. In order to do the *mapping-based* extraction Bizer et al. [2009] organize the infobox templates into a hierarchy, thus creating the DBpedia ontology with infobox templates as classes. They manually construct a set of property and object extraction rules based on the infobox class. This classification is more consistent as compared to the one obtained by means of generic extraction, however it has smaller coverage.

Other kinds of Wikipedia markup contributed to the DBpedia content as well. For example, values of the `rdfs:label` property are extracted from the human-readable representations of the Wikipedia page names, and first paragraphs of articles become the values of the `dbpedia:abstract`[34] property. Wikipedia category taxonomy is represented in DBpedia in terms of SKOS vocabulary with each Wikipedia category regarded as a SKOS concept. DBpedia "things" are then connected to the corresponding categories by means of the `http://purl.org/dc/elements/1.1/subject` predicate.[35] In addition to Wikipedia category information and DBpedia ontology classes, each "thing" is also classified in terms of YAGO categories (combination of Wikipedia categories with WordNet taxonomy, see Section 2.5.3) and UMBEL[36], a lightweight ontology intended for describing things on web. Recently, these classifications have been extended with the WordNet classification[37] created by manually mapping infoboxes into

[34] `dbpedia:` stands for `http://dbpedia.org/property/`

[35] "The topic of the resource" in terms of the Dublin Core Metadata Initiative (DCMI), `http://dublincore.org/documents/dcmi-terms/`

[36] http://umbel.org/

[37] `http://wiki.dbpedia.org/Datasets`

WordNet synsets.

As the Figure 2.3 shows, DBpedia is heavily interlinked with other datasets. It is connected by `owl:sameAs` links to at least 35 other datasets,[38] including Freebase, YAGO and OpenCyc.

DBpedia can be downloaded as a dump,[39] or it can be queried through a SPARQL endpoint, `http://dbpedia.org/sparql/`.

2.5.3 YAGO

YAGO [Suchanek et al., 2007] is an automatically created ontology. Its taxonomy is derived from WordNet and Wikipedia, and knowledge about individuals is extracted from Wikipedia.

The YAGO class taxonomy is created as follows. Suchanek et al. [2007] remove individuals from the WordNet taxonomy and convert the remaining data to the YAGO class system. Then they determine a subset of Wikipedia categories which they call *conceptual*. These categories identify entity classes, e.g. *English novelists*, in contrast to the other categories which define topics, e.g. *St John's College, Cambridge*, or are purely administrative. Conceptual categories are detected by using a heuristics which takes into account whether the head of a category is plural. Leaf conceptual categories are added as subclasses to the classes derived from the WordNet taxonomy by means of a heuristic algorithm, described in detail in [Suchanek et al., 2007]. In brief, Suchanek et al. [2007] parse the category names and map the obtained constituents to the WordNet synsets using the most frequent sense strategy. Pages that belong to conceptual categories become YAGO individuals.

Non-taxonomy relations are obtained by means of a variety of heuristics. `rdfs:label` relations are extracted from WordNet synonymy information

[38] `http://wiki.dbpedia.org/Downloads38#h236-1`

[39] `http://wiki.dbpedia.org/Downloads37?show_files=1`

and Wikipedia redirection links. Certain relations are extracted by applying patterns to the category names, e.g. if a page p belongs to category `<NN>_deaths` this would result in the fact $(p, diedIn, NN)$. More recently YAGO has been enriched with temporal and spatial dimensions [Hoffart et al., 2012], the latter is imported from GeoNames.[40] Currently, the core version of YAGO contains knowledge about 2.6 million entities and 124 million facts about them.[41]

The quality of YAGO has been assessed manually. Humans found a randomly selected subset of YAGO facts to be correct in 95% of the cases.

Yago can be downloaded as a dump.[42] YAGO SPARQL endpoint is hosted by OpenLink Software[43] as a part of the *OpenLink Software LOD Cache*.[44]

2.5.4 Freebase

Freebase [Bollacker et al., 2008] is a collaboratively constructed database originally developed by Metaweb and now owned by Google.[45] It contains knowledge automatically extracted from a number of resources, including Wikipedia, MusicBrainz,[46] NNDB,[47] Food and Drug Administration, and others.[48] The knowledge is supplied by both the automated data pipelines and the human volunteers.

Freebase can be considered as a huge graph. Its nodes have types "/type/object" and a set of narrower types. They are interconnected by

[40] `http://www.geonames.org/`

[41] `http://www.mpi-inf.mpg.de/yago-naga/yago/downloads.html`. Core version contains information only about entities present in Wikipedia/WordNet, without information about inner links from Wikipedia or GeoName entities that cannot be mapped to any Wikipedia article.

[42] `http://www.mpi-inf.mpg.de/yago-naga/yago/downloads.html`

[43] `http://www.openlinksw.com/`

[44] `http://lod.openlinksw.com/sparql`

[45] `http://googleblog.blogspot.it/2010/07/deeper-understanding-with-metaweb.html`

[46] `http://musicbrainz.org/`

[47] `http://www.nndb.com/`

[48] Full list available at `http://sources.freebaseapps.com/`

the edges corresponding to the node properties. Type of nodes define which properties they might have. Nodes typically correspond to the Freebase *topics*[49] which have the similar meaning to "things" in DBpedia, i.e. they can be concepts or an individual entities. Currently Freebase describes more than 23 million of them. Each topic is assigned a global unique identifier and a set of human-readable unique IDs, assembled of a key and a namespace. For example, one of the namespaces is the Wikipedia namespace, and respective key is the name of the Wikipedia page describing the topic. Moreover, topics are connected by means of `owl:sameAs` links to DBpedia.

Freebase data can be queried automatically in several ways. Its native query language is Metaweb Query Language (MQL). One can use an API to send automatic queries or run queries directly in the query editor.[50] In 2008 Metaweb created an RDF version of Freebase, thus making it part of the LOD cloud.[51]

2.5.5 WordNet

WordNet [Fellbaum et al., 1998] is a manually elaborated lexical semantic database developed in Princeton University. It organizes knowledge about the word meanings into a network of synsets. Synset is a collection of synonyms. Synsets are interconnected by various lexical relations, including hyperonymy, hyponymy, antonymy. Currently, WordNet is one of the most widely used resources in NLP.

Two WordNet versions are available on LOD: **WordNet (W3C)**, [Van Assem et al., 2006] converted from Princeton 2.0 WordNet Prolog distribu-

[49] Freebase can contain also nodes that are not topics, e.g. image metadata, see `http://wiki.freebase.com/wiki/Topic` for a complete list

[50] `http://www.freebase.com/queryeditor`

[51] `http://rdf.freebase.com/`

tion, and **WordNet (VUA)**,[52] an RDF verson of WordNet 3.0 created using the methodology similar to the one described in [Van Assem et al., 2006].

2.5.6 Cyc

Cyc is a comprehensive manually constructed knowledge base developed since 1984 by CycCorp. According to Lenat [1995] it can be considered as an expert system with domain spanning all everyday actions and entities, e.g. *Fish live in water*. Its development has taken more than 900 person-years [Matuszek et al., 2006]. Complete Cyc knowledge base contains more than 500,000 concepts and more than 5 million assertions about them. They may refer both to common human knowledge like food or drinks and to specialized knowledge in domains like physics or chemistry. A Cyc constant represents a thing or a concept in the world. It may be an individual, e.g. *BarackObama*, or a collection, e.g. *Gun*, *Screaming*.

Cyc is a proprietary commercial resource, however its full content is freely available for the research community as ResearchCyc. Originally, the knowledge base has been formulated using CycL language. In 2008, the open-source version of Cyc named OpenCyc,[53] which contains the full Cyc ontology and a restricted number of assertions, was made freely available as a part of LOD. A number of efforts connected Cyc to the other datasets. For example, OpenCyc concepts have been automatically linked to Wikipedia articles by Medelyan and Legg [2008] and Sarjant et al. [2009] with the purpose of further extending Cyc with Wikipedia knowledge such as new synonyms and translations. In addition, OpenCyc also contains `owl:sameAs` links to DBpedia, UMBEL, WordNet and other resources.

[52] `http://semanticweb.cs.vu.nl/lod/wn30/`
[53] `http://www.opencyc.org/`

Chapter 3

The Framework Implementation

In this chapter we provide the implementation details of the LOD-based semantic feature extraction: we describe the tool that we employ to link terms to Wikipedia and provide high-level details of the LOD knowledge extraction process.

3.1 Introduction

The main focus of this thesis is to exploit the Linked Open Data (LOD) datasets as a source of semantic knowledge in NLP, to detect the problems and to give practical solutions. We have already defined the high-level view of the framework for injecting semantic features to NLP in Section 1.3. In brief, it includes the following conceptual modules: (1) mapping terms in text to DBpedia URIs using Wikipedia as a mediator; (2) using the URIs to extract relevant knowledge from LOD; (3) extracting task-relevant features to be plugged into NLP engines. In this chapter we describe the practical details of how we realized these conceptual modules when performing our case studies.

The section is structured as follows. First, we describe the main components of our framework implementation and their interactions in Section 3.2. Then, in Section 3.3, we describe *The Wiki Machine* (TWM), the

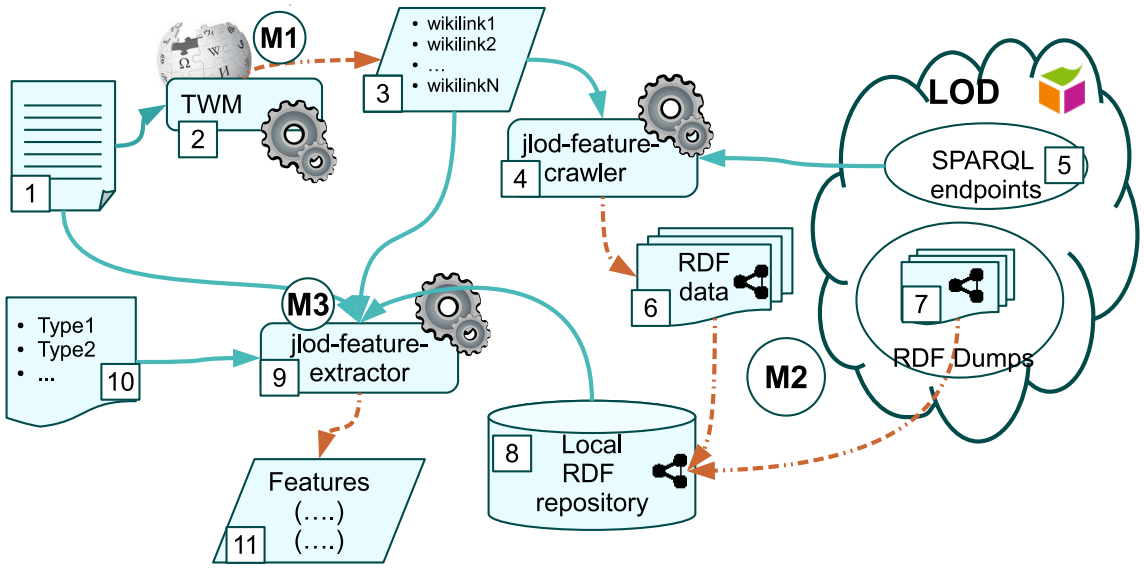

Figure 3.1: Overall schema of extracting LOD knowledge and converting it to NLP features.

tool which maps ambiguous terms to Wikipedia pages (module 1). Finally, in Section 3.4 we describe the technical details of how we extracted RDF data from LOD (module 2) and converted it into NLP semantic features (module 3). We have implemented the RDF data extraction and the feature extraction processes in the Java tool, *jlod-feature*,[1] which we plan to release as open-source.

3.2 Overall picture

We depict the high-level view of the framework in Figure 3.1. We have numbered the main components of the figure, and we will refer to them in our descriptions by specifying the component number in square brackets.

Module 1 (M1). We start feature extraction by processing the *original document [1]* with a *tool [2]* which annotates plain text with *Wikipedia*

[1]Planned to be available at `https://code.google.com/p/jlodfeat/`

links or DBpedia URIs [3] (Note that Wikipedia links are equivalent to DBpedia URIs). In our instantiation of the framework we use TWM (see Section 3.3) as [2].

Module 2 (M2). Then we populate the local *RDF repository [8]* with *knowledge about URIs of interest [6,7]* relevant for the NLP feature extraction. Theoretically, we could avoid having a *local repository [8]* and extract features on-fly, but this is time-consuming, because we have to send multiple queries to remote resources that might have different response times or can even be down for maintenance. *RDF repository [8]* can be populated in several ways which we describe below.

Local download. First, we can simply download the *RDF dumps [7]* of LOD datasets which we deem to be useful for a specific task and upload them into a local repository. This would always be reasonable if the dataset is not large-scale, e.g. an upper-level ontology or a type system, or if we are releasing an NLP tool. However, if are running some preliminary NLP experiments, it would be time- and space-consuming to download full versions of all the potentially relevant large-scale LOD datasets.

Crawling. The second way is to crawl LOD by dereferencing the URIs. If, for example, we have dereferenced a URI corresponding to a term in a plain text and retrieved a corresponding RDF graph, we might require some specific information about the other URIs in this graph. Then we might have to dereference those URIs as well and lose time for waiting for a response from a distant server. Moreover, we may retrieve information that is abundant for our purposes since dereferencing returns *all* information deemed to be interesting by the data provider.

SPARQL endpoints. Finally, we can employ *SPARQL endpoints [5]* to extract, when available, to extract portion of LOD knowledge relevant for our NLP task.

In our case-studies we used the first and the last options, that is we downloaded some resources locally (e.g. OpenCyc), and sent SPARQL queries to the other resources. In Section 3.4.1 we describe our SPARQL query process. The process is implemented in the *jlod-feature-crawler package [4]* of the *jlod-feature* tool.

Module 3 (M3). Finally, we pass the list of *URIs of interest [2]*, a *local RDF repository [8]* containing useful knowledge, the *original terms of interest [1]*, and a *list of feature types [9]* that we need to extract to the *feature extraction tool [10]*. Our feature extraction tool is a module of *jlod-feature* called *jlod-feature-extractor*. We describe the feature types and the tool in more detail in Section 3.4.2. The output of the process is a *feature representation file [11]* to be used by a machine learning algorithm.

3.3 The Wiki Machine

First step of our framework consists in annotating terms in plain text with links to Wikipedia pages. Wikipedia, along with its structured representation DBpedia, is heavily interlinked with LOD datasets. Therefore, it can be used for linking terms in plain text to URIs in LOD datasets.

We have used the tool called *The Wiki Machine (TWM)*.[2] TWM is a supervised kernel-based word sense disambiguation system employing local and global context clues. The approach is summarized in the following subsections.

[2] `http://thewikimachine.fbk.eu/`

3.3.1 Training set

Training data is automatically extracted from Wikipedia as it was first proposed in [Cucerzan, 2007, Mihalcea, 2007]. To create the training set, for each term of interest m, TWM collects from the English Wikipedia dump[3] all contexts where m is an anchor of an internal link, where a context corresponds to a line of text in the Wikipedia dump and is represented as a paragraph in a Wikipedia article. The set of target articles represents the senses of m in Wikipedia and the contexts are used as labeled training examples. E.g., the proper noun *Bush* is a link anchor in 17,067 different contexts that point to 20 different Wikipedia pages, `George_W._Bush`, `Bush_(band)`, and `Dave_Bush` are some examples of possible senses. The set of contexts with their corresponding senses is then used to train the WSD system described below. E.g., the context "*Alternative Rock bands from the mid-90 's , including Bush , Silverchair , and Sponge.*" is a training instance for the sense defined by the Wikipedia entry `Bush_(band)`.

3.3.2 Learning algorithm

To disambiguate terms in text, TWM employs a kernel-based approach originally proposed in [Giuliano et al., 2009]. Different kernel functions are employed to integrate syntactic, semantic, and pragmatic knowledge sources typically used in the WSD literature. Kernel methods are theoretically well founded in statistical learning theory and have shown good empirical results in many applications [Shawe-Taylor and Cristianini, 2004]. Their strategy adopted consists in splitting the learning problem into two parts. They first embed the input data in a suitable feature space, and then use a linear algorithm (e.g., support vector machines) to discover nonlinear patterns in the input space. The kernel function is the only task-specific

[3] `http://download.wikimedia.org/enwiki/20100312` for experiments in Chapter 4, and `http://download.wikimedia.org/enwiki/20120601` in Chapter 5

component of the learning algorithm. For each knowledge source a specific kernel has been defined. By exploiting the property of kernels, basic kernels are then combined to define the WSD kernel. Specifically, TWM is based on a linear combination of gap-weighted subsequence, bag-of-words, and latent semantic kernels.

Gap-weighted subsequences kernel. This kernel learns syntactic and associative relations between words in a local context. Roughly speaking, it compares two sequences of words by means of the number of contiguous and non-contiguous sequences of a given length they have in common. The kernel employed by TWM is extended with subsequences of word forms, stems, part-of-speech tags, and orthographic features (capitalization, punctuation, numerals, etc.). Gap-weighted subsequences kernels employed in TWM work on subsequences of length up to 5. E.g., suppose one needs to disambiguate the verb "to score" in the context "Maradona scored Argentina's third goal", given the labeled example "Ronaldo scored two goals in the second half" as training, a traditional approach, that only considers contiguous ngrams, has no clues to return the correct answer because the two contexts have no features in common. The use of gap-weighted subsequences allows to overcame this problem and extract the feature "score goal," shared by the two examples.

Bag-of-words kernel. This kernel learns domain, semantic, and topical information. Bag-of-words kernel takes as input a wide context window around the target mention. Words are represented using stems. The main drawback of this approach is the need of a large amount of training data to reliably estimate model parameters. E.g., despite the fact that the examples "People affected by AIDS" and "HIV is a virus" express related concepts, their similarity is zero using the bag-of-words model since they

have no words in common (they are represented by orthogonal vectors). On the other hand, due to the ambiguity of the word "virus", the similarity between the contexts "the laptop has been infected by a virus" and "HIV is a virus" is greater than zero, even though they convey very different messages.

Latent semantic kernel. Latent semantic kernel helps to overcome the drawback of the bag-of-words. It incorporates semantic information acquired from English Wikipedia. This kernel extracts semantic information through co-occurrence analysis in the corpus. The technique used to extract the co-occurrence statistics relies on a singular value decomposition (SVD) of the term-by-document matrix. E.g., the similarity in the latent semantic space of the two examples "People affected by AIDS" and "HIV is a virus" is higher than in the bag-of-words representation, because the terms AIDS, HIV and virus very often co-occur in the medicine domain.

3.3.3 Implementation details

The TWM latent semantic model is derived from the 200,000 most visited Wikipedia articles. After removing terms that occur less than 5 times, the co-occurrence matrix contains about 300,000 and 150,000 terms respectively. TWM uses the SVDLIBC package to compute the SVD, truncated to 400 dimensions.[4] To classify each mention in Wikipedia entries, TWM uses a LIBSVM package.[5] No parameter optimization is performed.

3.3.4 Evaluation

We have evaluated TWM on the ACE05-WIKI Extension [Bentivogli et al., 2010]. This dataset extends the English Automatic Content Extraction

[4] http://tedlab.mit.edu/~dr/svdlibc/

[5] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

(ACE) 2005 dataset with ground-truth links to Wikipedia.[6] ACE 2005 is composed of 599 articles assembled from a variety of sources selected from broadcast news programs, newspapers, newswire reports, internet sources and from transcribed audio. It contains the annotation of a series of entities (person, location, organization) and their mentions. In the extension each nominal or named entity mention (in total 29,300 entity mentions) is manually assigned a Wikipedia link(s). If a mention is assigned more than one link, the links are ordered from more specific to less specific. The results of the evaluation are reported in the second part of Table 3.1. The evaluation is performed considering only the most specific ACE05-WIKI links as gold standard annotations.

We have compared our approach with the state-of-the-art system as for 2010, *Wikipedia Miner* tool[7] [Milne and Witten, 2008]. We used it with the default parameters. The tool requires a Wikipedia dump preprocessed in a special way. We used the preprocessed Wikipedia dump of July, 2008, made available by the authors of the tool. The results are reported in the first part of Table 3.1. The Wikipedia Miner achieves six points better precision, however, its recall is considerably lower, thus making the F_1 12 points less than that of TWM.

The performance difference between the two systems could not be only due to the use of different versions of Wikipedia, as the ACE corpus contains references to entities dated before 2005 and Wikipedia covered most of them in 2008. On the other hand, varying the Wiki Miner free parameters did not produce significant improvement.

Mendes et al. [2011] compared TWM to the other similar tools and observed it to have the highest F_1-measure as compared to the other systems, including DBpedia Spotlight [Mendes et al., 2011], Zemanta[8] and

[6] http://www.itl.nist.gov/iad/mig//tests/ace/ace05/index.html
[7] http://wikipedia-miner.sourceforge.net/
[8] http://www.zemanta.com/demo/

Approach	Mention Type	Precision	Recall	$\mathbf{F_1}$
Wikipedia Miner	NAM & NOM	$0.78^{\dagger}$	0.48	0.59
	NAM	$0.86^{\dagger}$	0.69	0.76
	NOM	0.66	0.28	0.40
The Wiki Machine	NAM & NOM	0.72	$0.71^{\dagger}$	$0.71^{\dagger}$
	NAM	0.78	$0.74^{\dagger}$	0.76
	NOM	0.62	$0.65^{\dagger}$	$0.63^{\dagger}$

Table 3.1: Comparative evaluation of the two disambiguation methods on ACE05-WIKI (micro-average). Symbol † indicates significant differences relative to the corresponding mention type ($p < 0.01$). Significance tests are computed using approximate randomization procedure.

OpenCalais.[9]

3.3.5 Related work

Formally, the task of linking to Wikipedia can be formulated as follows: given a document d and a set of terms of interest t_i in it, $(i = 1, N)$, one needs to annotate each t_i either with a link to a Wikipedia page w_i describing its meaning in d or to specify that no such page exists. This is a word sense disambiguation (WSD) task.

Special cases of linking to Wikipedia include named entity disambiguation, if t_i are named entity mentions; *wikification*, if t_i are terms important for understanding d; or a knowledge base entity linking problem, if t_i-s are the potential mentions of the knowledge base elements. The latter has been payed special attention in the Knowledge Base Population (KBP) task [Ji et al., 2010, McNamee and Dang, 2009, Ji et al., 2011] of the Text Analysis Conference (TAC).[10] One of KBP substasks consists in linking entity mentions in a document to a knowledge base with information about entities

[9] http://www.opencalais.com/

[10] http://www.nist.gov/tac/

corresponding to Wikipedia articles.

The task of linking to Wikipedia typically consists in (i) building a term-sense dictionary, D, where senses are Wikipedia articles; (ii) devising a methodology, M, to select an appropriate sense for a term in context. Typically, terms which may denote a specific sense (Wikipedia page), w, in D are collected from page titles, titles of redirection and disambiguation pages, and anchor texts of the hyperlinks pointing to w [Csomai and Mihalcea, 2008, Bunescu and Pasca, 2006, Milne and Witten, 2008]. Some approaches employ more sophisticated techniques, e.g. collecting web search queries that result in a click on a link to w [Zhou et al., 2010], or cast the term-to-article mapping task as a machine translation problem [Han and Sun, 2011]. As for M, the disambiguation methodologies typically follow the intuition that the following phenomena are valuable when determining if a candidate page, w_c, is a correct assignment for t_i: (i) similarity between contexts of t_i in d and the Wikipedia contexts[11] of w_c; (ii) topical coherence and semantic relatedness of the Wikipedia pages assigned to all t_i-s; (iii) prior probability of w_c to be a sense of t_i. Ratinov et al. [2011] named (i) the *local* and (ii) the *global* evidences.

For example, Bunescu and Pasca [2006] performed local disambiguation, taking into account cosine similarity of the t_i's context in d to the text of w_c and the correlation between words in d and the categories of w_c. Similarly, Mihalcea [2007], Csomai and Mihalcea [2008] used Wikipedia articles as a sense-tagged corpus, where sense tags are the target pages, to train a supervised data-driven WSD classifier.

Later approaches typically combined local and global evidences. Cucerzan [2007] tackles Named Entity Disambiguation (NED), his t_i are named entity mentions. In order to disambiguate t_i, he optimizes the function that incorporates similarity of Wikipedia contexts of w_c to t_i's contexts in d and

[11] For example, text of w_c or words around the links pointing to w_c from the other pages

w_c's topical coherence with all the disambiguations of all t_j ($j \neq i$) in d. Coherence is measured using the Wikipedia category structure.

Milne and Witten [2008] tackled the task of wikification relying on Wikipedia link network only. They used Wikipedia pages that can be unambiguously linked to terms in d to form a *page context*. Probability of w_c to be a correct disambiguation for t_i was evaluated based on *commonness*, *relatedness* and *quality of context*. Here *commonness* is prior likelihood of t_i to be linked to w_c estimated on the Wikipedia link structure; *relatedness* is relatedness of w_c to the pages in the page context measured by means of relatedness function defined on Wikipedia internal links [Milne and Witten, 2009]; and *quality of context* reflects relatedness of context pages to each other. Their approach does not require such extensive text preprocessing as those using text-based similarity and shows competitive results. However, its limitation is that it relies on presence of non-ambiguous terms in the context, which is not always the case.

Ferragina and Scaiella [2010] pointed that the approach by Milne and Witten [2008] might encounter problems when processing short texts due to the lack of monosemous context terms in them. Instead of using relatedness to monosemous context, they introduce a "collective agreement" function. It takes into account all candidate senses of all context terms and their input weighted by their commonness. Ratinov et al. [2011] proposed another procedure for the page context population. First, they disambiguate terms based on similarity of t_i's contexts to Wikipedia contexts of candidate pages. They used the obtained disambiguations to populate the page context. Then, the page context is used in their method which combines local and global evidence.

Kulkarni et al. [2009] optimize the function incorporating local and global evidence, taking into account all possible senses of all t_i-s to form the global evidence. They annotate all entity mentions not one-by-one

but jointly. The function employed by their method is NP-hard, so they propose an approximation.

Currently, there exist a number of APIs that perform linking to Wikipedia or DBpedia. Non-commercial APIs include Wikipedia Miner [Milne and Witten, 2009, 2008],[12] TagMe [Ferragina and Scaiella, 2010],[13] DBpedia spotlight [Mendes et al., 2011].[14] Commercial tools include Zemanta,[15], AlchemyApi[16] and OpenCalais.[17]

3.4 LOD-based semantic feature extraction schema and implementation details

3.4.1 Extracting LOD data relevant for feature extraction

When developing an NLP application one typically runs a set of experiments on a corpus in order to define the best algorithm and feature configuration. When running experiments, we do not know in advance which LOD sources might constitute a valuable source of features. Fully downloading all possibly useful large-scale sources such as DBpedia or Freebase locally takes space and time and can slow down the experiments. It is more reasonable to download portions of task-relevant knowledge about the URIs in the specific corpus.

If we do not want to download a complete dump of some LOD resource and this resource is accessible through a SPARQL endpoint, we proceed as follows. Given a DBpedia URI, `<dburi>`, we query the endpoint for the URIs connected to it by means of `owl:sameAs` predicate. Then, for each retrieved URI, `sameAsURI`, and the original `<dburi>` we query the endpoint

[12] http://wikipedia-miner.cms.waikato.ac.nz/
[13] http://tagme.di.unipi.it/
[14] https://github.com/dbpedia-spotlight/dbpedia-spotlight
[15] http://www.zemanta.com/
[16] http://www.alchemyapi.com/
[17] http://www.opencalais.com/

for all the RDF triples matching the pattern *(<dburi OR sameAsURI> ?p ?o)*. Here *?p* and *?o* are the variables to be bound. Then we recursively repeat the same query n times for each binding of *?o*. However, when repeating the queries for the *?o* bindings we impose a filter on *?p*, requiring the URI of the latter to partially match a list of manually selected keywords such as "subject", "type", "class", "label" etc. We have implemented this procedure in the *jlod-feature-crawler* package of the *jlod-feature* tool.

The motivation behind such query strategy is the following. We believe that the most beneficial information for our purposes are the RDF statements which have a URI corresponding to the term of interest as subject, i.e. statements describing direct *properties* of a concept referred to by URI. Another useful kind of information is hierarchical type and topic information about a given URI. For example, direct properties of objects can include their aliases, knowledge useful in coreference resolution (see Section 5.2). For instance, `fb:ibm`[18] is the subject of the triple with predicate `fb:common.topic.alias` and object "Big Blue". Hierarchical (i.e. type, type generalization or hyperonymy) type information is useful in both relation extraction and coreference resolution (see Section 4.2 and Section 5.2), and in a number of other tasks, including textual entailment or information retrieval. For example, taxonomic knowledge that `dbpedia:IBM` has type `dbpedia-owl:Company` that is a subclass of `dbpedia-owl:Organization` is relevant for relation extraction, coreference resolution or textual entailment.

3.4.2 Extracting features

In this section we describe the features currently extracted by the *jlod-feature-extractor* module of the *jlod-feature* tool. We believe that these kinds of features are universally useful for NLP tasks.

[18] `fb:` corresponds to `http://rdf.freebase.som/rdf/`

Term-level features. Extracted for each term of interest in a document/corpus separately. They reflect information about the types of the "things" identified by URI assigned to a term of interest, their generalizations or topics. Currently *jlod-feature-extractor* contains term-level feature extractors for the WordNet VUA 3.0, YAGO, OpenCyc and DBpedia data schemas. The features include hyperonymy information from WordNet, class generalization information from OpenCyc and YAGO. For instance, term-level generalization features of *Batallion* extracted from OpenCyc include *MilitaryOrganization Group*. We have employed the term-level features in the semantic relation extraction experiments in Chapter 5. See the Section 5.3.3 for the detailed feature descriptions.

Term-pair level features. Features of this type are extracted for pairs of terms of interest, t_1 and t_2, annotated with URI_1 and URI_2, respectively. Given a URI_i $(i = 1, 2)$ we extract a subgraph g_i from a given RDF data repository that meets the following requirements: g_i contains URI_i, and the maximal distance of all its nodes to the node corresponding to URI_i is less than n edges. We union the graphs g_1 and g_2 into a graph G.

We can use these structures, namely g_1, g_2 and G, to extract features indicating the connection between "things" referred to by t_1 and t_2. For example, in *jlod-feature-extractor* we have implemented the feature extractors that extract features from paths between URI_1 and URI_2 in G. For instance, if URI_1 =`dbpedia:MSNBC` and URI_2 =`dbpedia:Television_network` one of the paths connecting them in the union of DBpedia and OpenCyc is

`dbpedia:MSNBC`[19] → `rdf:type` → `opencyc:Mx4rvjMrW5wpEbGdrcN5Y29ycA`[20]
→ `rdfs:subClassOf` → `opencyc:Mx4rwQCRtJwpEbGdrcN5Y29ycA`
→ `owl:sameAs` → `dbpedia:Television_network`

[19] `dbpedia:` means `http://dbpedia.org/resource/`
[20] `opencyc:` means `http://sw.opencyc.org/concept/`

From such path *jlod-feature-extractor* extracts the feature that lists all predicates connecting two URIs. In this specific case it is `rdf:type`→`rdf:subClassOf`.

Additionally, *jlod-feature-extractor* extracts features that indicate whether we observe full or partial string match between t_2 and literal or URI nodes in g_1, and vice versa. We employed the partial string match features with $n = 1$ for the coreference resolution experiments in Chapter 4.

For example, if URI_1=`dbpedia:MSNBC`, then g_1 extracted from DBpedia includes the following statement:

> (subject = `dbpedia:MSNBC`, predicate = `rdf:type`, object = `dbpedia-owl:Organisation`)[21]

Then if t_2="organization" we can extract the feature `partialStringMatch_rdf:type`.

[21] `dbpedia-owl` means `http://dbpedia.org/ontology/`

Chapter 4

Coreference resolution

In this chapter we report the case-study of applying our framework for the task of coreference resolution. We annotated entity mentions with links to Wikipedia, used the links to extract RDF-encoded knowledge from Linked Open Data sources, namely YAGO, Freebase, and DBpedia, and applied feature selection techniques to extract the relevant subset of semantic features. We incorporate the new features into a baseline coreference resolution system implemented as Markov Logic Network. By means of experiments on ACE 2005 corpus we show that background knowledge helps to increase the overall MUC F_1 measure, due to the increase in recall.

4.1 Introduction

The task of noun phrase (NP) coreference resolution consists in identifying which noun phrases and pronouns in a text, called mentions (also called *references* or *markables*), refer to the same entity. For example, resolving coreference means identifying that the mentions *Barack Obama*, *president* and *he* in the text *"**Barack Obama** will make an appearance on the TV show, the **president** is scheduled to come on Friday evening, and **he** is*

expected to talk about health-care issues." refer to same real-world entity. This constitutes an important subtask in many NLP tasks, such as information extraction, textual entailment, and question answering. The task of coreference resolution is a complex task, and it can be split into multiple subtasks. First, one needs to detect entity mentions in plain text, then choose a machine learning technique (choose a machine learning algorithm, an approach to obtain a balanced set of negative and positive instances if the algorithm is supervised, a clustering technique for multiple entity mentions). Finally, another important component is selecting linguistic or commonsense intuitions of which clues might be indicative of coreference and encoding these intuitions as features or rules.

The main focus of our work in this chapter is the latter subtask. We follow the intuition that semantic knowledge, including encyclopedic or common-sense knowledge, can be helpful when resolving coreference. For example, knowledge that *Barack Obama* is a *president* is useful in the case of the example provided above. So far the majority of approaches extracted semantic knowledge from WordNet [Soon et al., 2001], gazeeteers [Bengtson and Roth, 2008], output of Named Entity Recognition systems [Ng, 2007], and, more recently, Wikipedia [Ponzetto and Strube, 2006]. The problem is that some of these sources, e.g. WordNet might be limited in coverage, especially for the named entity mentions, while others might have noisy structure, e.g. Wikipedia. Another problem that emerges when selecting relevant knowledge for a given entity mention from an external source is the ambiguity of natural language.

In this chapter we extract semantic knowledge from the Linked Open Data (LOD) datasets (See Chapter 2). First, LOD is assembled of a large number of large-scale resources, therefore it is unlikely to suffer from the problem of coverage. Second, unlike Wikipedia, LOD resources are formally structured, thus knowledge extracted from them is less likely to be

noisy. Finally, many of them are aligned with Wikipedia. This allows us to use a Wikipedia-based word sense disambiguation system for mapping terms to LOD URIs. These considerations motivated our investigation, in which we integrate the LOD knowledge into the coreference resolution task by employing our framework, described in Chapter 3.

Following the outline of the framework, we first map entity mentions to Wikipedia, using *The Wiki Machine* (TWM), a supervised word sense disambiguation system (see Section 3.3). Then, we use Wikipedia link as a semantic mediator to obtain background knowledge about entity mentions from Freebase, YAGO and DBpedia (Section 2.5), sources selected due to their high coverage on common nominals and named entities. We convert the obtained knowledge into features, and run a feature selection algorithm to detect the relevant feature subset. Finally, we add features to a baseline coreference resolution system, implemented as a Markov Logic Network [Domingos et al., 2008] and run experiments in proper and common noun coreference resolution. We show that the new semantic features are beneficial for resolving the coreferences between proper and common noun mentions.

This chapter is structured as follows. First, we overview the related work (Section 4.2). Then we describe how we extracted features and selected their subset relevant for coreference resolution (Section 4.3). Finally, we inject the selected features into a coreference resolution framework, and report a set of experiments on ACE 2005 English corpus (Section 4.4).

4.2 Related work

The early works on the subject tackle the task of anaphora resolution. It is closely related, although not fully equivalent to the coreference resolution.

Anaphora is a reference, referent[1] of which cannot be identified without supplementary information, and depends on another reference in a text, called *antecedent*. For instance, in the example given in the introduction *he* is an anaphora, while *Barack Obama* is an antecedent.

First approaches to the task of anaphora resolution were rule-based, and employed syntactic intuitions [Hobbs, 1978], discourse centering theory [Grosz et al., 1995], common-sense reasoning [Winograd, 1972, Wilks, 1975] or their combination [Carter, 1987]. Starting from the 90's the substantial effort required for manual encoding of the rules caused the research to shift towards the empirical machine learning approaches, which are now considered as state-of-the art in the coreference task.

The task of coreference resolution in the context of information extraction was introduced in the sixth Message Understanding Conference (MUC) competition [Grishman and Sundheim, 1996] organized by the Defense Advanced Research Projects Agency (DARPA). The committee explained their decision, saying that identifying coreferent expressions would result in deeper understanding of text by automated information extraction systems. Since then it has been a sub-part of the following evaluation campaigns, including MUC-7 [Chinchor and Hirschmann, 1997] and Automatic Content Extraction (ACE) evaluation campaigns [ACE, 2000-2005].

Modern state-of-the-art coreference resolvers are mostly extensions of the approach by Soon et al. [2001] in which a mention-pair classifier is trained using a set of twelve surface-level features. A mention-pair classifier first classifies pairs of entity-mentions as either coreferent or not, and then clusters the coreferent pairs into coreference chains of entity mentions referring to the same object.

In the last decade, two independent research lines have extended the Soon et al. approach yielding significant improvements in accuracy. First

[1] entity it refers to

line concerns improving the machine learning algorithms, including strategies for creating training/testing instances and creating the coreference chains (see Section 4.2.1). The second line is concerned with increasing the amount of features indicative of the coreference, paying special attention to the semantic-based features (see Section 4.2.2).

4.2.1 Machine learning approaches to coreference resolution

The ML methods for coreference can be classified by the technique employed to create clusters of coreferring mentions, i.e. coreference chains, approaches to generate positive/negative train/test instances, and the degree of supervision required by the algorithm.

As stated above, one of the first techniques (or models) for creation of coreference chains is the mention-pair model, employed in [Soon et al., 2001]. However, this model has been criticized for lacking expressiveness. Ng [2010] uses the following example, in order to demonstrate limitations of this model. Given a list of mentions "Barack Obama", "Obama" and "she", the mention pair model might extract two coreferring pairs, ("Barack Obama", "Obama") and ("Obama", "she"). These pairs will be further merged into a single cluster ("Barack Obama", "Obama", "she"), where "she" is incompatible with "Barack Obama".

Entity-mention and mention-ranking models and their combination cluster-ranking are some of the relevant approaches proposed to avoid the problems posed by the mention-pair models (e.g. Denis and Baldridge [2007], Ng [2004]). In entity-mention models an entity mention is checked for co-reference with a cluster of entity-mentions classified as co-referring. In the aforemention example "she" would be checked for coreference with the cluster ("Barack Obama", "Obama"), and the gender disagreement will indicate the absense of the co-reference.

The mention-ranking models, instead classifying of a mention pairs rank

all candidate antecedents of a mention of interest, and select the most highly-ranked one as the antecedent [Iida et al., 2003, Yang et al., 2003].

Another distinction between the ML models for coreference resolution is the degree of supervision they require. Initially, the unsupervised models were considerably outperformed by the supervised models, however recently combination of the entity-mention mention clustering model and unsupervised techniques show the results comparable to that of the supervised approaches [Haghighi and Klein, 2010]. Unsupervised co-reference resolution may be based on generative models [Haghighi and Klein, 2010], Markov logic networks [Poon and Domingos, 2008], and, more recently, on multi-sieve approaches [Raghunathan et al., 2010] that apply a set of models moving from high-precision models to the lower-precision ones.

In our work we do not aim to improve the algorithmic aspect of the coreference resolution framework, while aiming at improving the feature representation of the instances to be processed by it.

4.2.2 Semantic features employed for co-reference resolution

Features employed by coreference resolution systems encode various linguistic intuitions concerning this phenomena. Ng [2010] divides them into string-matching, syntactic, grammatical, discourse-based and semantic.

The semantic features typically reflect the intuition that semantic compatibility of entity mentions, their gender agreement and knowledge whether one of them is an alias of another may be indicative of coreference. In many cases such knowledge cannot be obtained directly from the text in consideration. Therefore, one of the major research lines in the recent years investigates the usage of semantic knowledge sources to augment the semantic feature space [Soon et al., 2001, Ponzetto and Strube, 2006, Ng, 2007, Versley et al., 2008]. Here the majority of the approaches exploit WordNet [Fellbaum et al., 1998], gazetteers and name lists, distributional

similarity, corpora annotated with semantic classes, and, more recently, Wikipedia[2].

WordNet and gazetteers. In one of the earliest machine learning approaches to coreference resolution, [Soon et al., 2001], a candidate pair of mentions (m_i, m_j) was represented as a vector of twelve features, where semantic features were represented by the alias and semantic class agreement features. Alias was a binary feature obtained using a set of heuristics, e.g. it was considered true if one mention was an acronym of another. Consequently, its value could be obtained only in a limited number of cases. In order to extract the semantic class agreement feature Soon et al. [2001] created a very coarse-grained set of semantic classes and mapped them to the corresponding WordNet synsets located at the top of the WordNet taxonomy. A semantic class for a mention was obtained my picking its most frequent WordNet sense and exploiting the WordNet hyponymy relations.

Experimental results showed that the alias feature contributed greatly to the performance of the system, while the semantic class compatibility had no impact. Authors point at the fact that their semantic class system might be too coarse-grained, and, moreover, the semantic class annotation was very noisy. This may be due to the absence of the word sense disambiguation [Ng, 2007]. Moreover, given that WordNet is assembled manually, it might lack coverage, especially for the named entity or domain-specific mentions.

Ng and Cardie [2002] aimed at incorporating more linguistic intuitions into the coreference resolution systems and expanded all the categories of features from [Soon et al., 2001]. The new semantic features were based on the WordNet ancestor/descendant relationship between entity mentions, and the distance between their heads in this hierarchy. As in [Soon et al.,

[2] http://wikipedia.org/

2001], mentions were mapped to WordNet without using word sense disambiguation. The extended feature set, including new semantic features, resulted in decrease of performance, and many of the features, including the WordNet-based features, were eliminated. The authors' overall intuition for this issue is that the full feature set might have been insufficient, or features' number was too large for their training set.

Both semantic and non-semantic features employed by Ng and Cardie [2002], Soon et al. [2001] became a "standard" baseline feature set widely employed in multiple other works, e.g. [Culotta et al., 2007, Yang and Su, 2007, Poesio et al., 2004]. Additionally, in order to increase the coverage for the case of proper nouns, some of the works employ combination of WordNet and gazetteers [Bengtson and Roth, 2008].

NER systems. Ng [2007] assumed that a possible reason why the semantic class agreement features did not contribute to the output of [Soon et al., 2001, Ng and Cardie, 2002], is the fact that they obtain semantic classes of entity mentions by mapping them to WordNet without disambiguation. Ng [2007] trains a semantic class induction system on the BBN corpus to annotate noun phrases with ACE semantic classes. WordNet information is used indirectly, as one of the features for the semantic class induction system. Ng [2007] shows that features or constraints based on the induced classes help to improve over [Soon et al., 2001], thus demonstrating that taking into account the ambiguity of mentions is crucial for obtaining the semantic knowledge relevant for coreference resolution.

The later work by Haghighi and Klein [2009] obtains semantic classes of entities using Stanford Named Entity Recognizer [Finkel et al., 2005] as an off-the-shelf tool. Semantic classes used by [Ng, 2007] and [Haghighi and Klein, 2009] are more accurate than those derived by means of WordNet in the earlier work, however due to their coarse-grained nature, they might

increase the number of false-positives.

Wikipedia. In addition to WordNet, gazetteers and NER systems, in the recent years Wikipedia became a frequently used source of semantic information. For example, Ponzetto and Strube [2006] introduce new features extracted from Wikipedia, WordNet and output of a semantic parser. The WordNet-based features consist in semantic similarity between entity mentions in a candidate coreference pair evaluated employing similarity measures defined on the WordNet taxonomy [Pedersen et al., 2004]. More specifically, they obtain a set of similarity scores for all possible pairwise combinations of WordNet senses to which the entity mentions heads can be mapped and then use the maximal score and the average of all scores as features.

In order to obtain Wikipedia features, Ponzetto and Strube [2006] map both entity mentions of interest to corresponding Wikipedia pages using a heuristic that is likely to return the most frequent sense. Features extracted for a coreference candidate pair include (i) gloss[3] overlap score of the pages obtained, (ii) their semantic relatedness calculated using Wikipedia category structure, and (iii) various partial string matches, e.g. match between one mention and the anchor text of the links, abstracts or categories of the page corresponding to the other mention. They observe that the new features helped to increase recall for common noun coreference resolution, thus resulting in high F_1 measure.

Ratinov and Roth [2012] also map entity mentions to Wikipedia,[4] using a supervised disambiguation system *GLOW* [Ratinov et al., 2011], that maps terms to Wikipedia taking into account their context. Categories of the pages (i) are used to extract the nationality of an entity mention,

[3]first paragraph of a Wikipedia page

[4]Note that their work [Ratinov and Roth, 2012] was published later than our work [Bryl et al., 2010]

(ii) are converted to fine-grained entity types by means of a heuristic algorithm. Moreover, the first paragraph of a page is employed to detect gender of a corresponding mention. The features extracted from this knowledge are used to extend the feature set by [Bengtson and Roth, 2008]. Our framework is similar in spirit but permits to extract knowledge directly for structured sources without need to resort to heuristics. The new features indicate whether two entities have been annotated with the same Wikipedia page, include nationality and gender agreement, and features based on the intersection of the sets of fine-grained semantic classes for both mentions. The combination of the knowledge-based features and the novel machine-learning framework provides an improvement over the baseline by [Bengtson and Roth, 2008].

Wikipedia has also been used as a corpus for mining patterns indicative of coreference [Yang and Su, 2007, Haghighi and Klein, 2009], extracting lists of semantically compatible word pairs [Haghighi and Klein, 2009], or training a generative unsupervised model [Haghighi and Klein, 2010]. Wikipedia list pages gave been used instead of gazetteers by Raghunathan et al. [2010]. They extract the lists of denonyms, e.g. *Italy-Italian*, from the Wikipedia list page dedicated to this subject.

Semantic role labels. Ponzetto and Strube [2006] exploited PropBank semantic roles label (SRL) annotations of entity mentions supplied by the ASSERT [Pradhan et al., 2004] semantic parser as features. They conclude that these features are beneficial for pronoun resolution.

Rahman and Ng [2011] combine semantic parse knowledge provided by the ASSERT parser with knowledge from FrameNet. FrameNet features indicate whether two mentions in consideration occur in the same frame, different frames, or whether at least of one of them does not occur at all. ASSERT-based features indicate whether combination of semantic roles of

the mentions falls into five predefined categories. Finally, they also created the joint FrameNet-ASSERT features, by combining the features described above. SRL features were shown to be useful when exploited in combination with other features, including information from YAGO, apposition and noun/verb pair compatibility obtained from labeled corpora.

Large-scale knowledge bases. To our knowledge we were first to apply YAGO for coreference resolution [Bryl et al., 2010].[5] The details of our approach are described in the following sections of this chapter.

Later, features from YAGO were exploited by Rahman and Ng [2011]. Consistently with us [Bryl et al., 2010] they use *means* and *type* relations from YAGO for the cases when one mention is a proper and another is a common noun, but employ more sophisticated machine-learning models for coreference resolution. They do not disambiguate. The experiments showed that the YAGO *type* feature was among the features with the largest performance gain in their system.

Lee et al. [2011] exploit Freebase and Wikipedia infoboxes in addition to WordNet. Their model has multi-sieved architecture, and the semantic sieve already assumes presence of some primary clustering of entity mentions. They map clusters to Freebase and Wikipedia, using the longest entity mention in a cluster that has a match, and employ the most frequent sense strategy in case of ambiguity. Freebase "name" and "alias" fields along with information from Wikipedia infoboxes are exploited as a source of alias information to be used in a newly introduced alias sieve.

[5]Note that the other works reported below were published later than our work [Bryl et al., 2010]

4.3 Background knowledge (BK) acquisition

In this section we describe how we extract background knowledge (BK) and select features relevant for coreference resolution.

First, we annotate all non-pronominal entity mentions with links to Wikipedia, using *The Wiki Machine* (TWM), described in Section 3.3. Then we use the Wikipedia links to extract RDF knowledge about entity mentions from LOD sources. The sources employed in this chapter are YAGO, DBpedia and Freebase. The amount of information obtained from a single LOD resource for a named entity can be very large. For instance, DBpedia alone contains around 600 RDF triples describing *Barack Obama*. Most of this information is irrelevant to the NLP task at hand (e.g. Obama's website, residence, the name of his spouse, etc.), and only some of the triples can be useful to resolve coreferences (e.g. `rdf:type` properties stating that Obama is a politician and a president).

Indeed, many learning algorithms are originally not designed to deal with large amounts of irrelevant information, consequently, combining them with the *feature selection* techniques has become necessary in many applications. This is particularly true when the information needed is retrieved from heterogeneous knowledge sources as the ones made available on the LOD.

We use the chi-square test to assess the relevance of background knowledge for the coreference resolution task by looking only at the intrinsic properties of the data. The chi-square test is a test for dependence between a feature and a class. Specifically, chi-square metric is calculated for each feature, and low-scoring features are removed. Afterwards, this subset of features is presented as input to the learning algorithm. Benefits of the chi-square test are that it easily scales to very high-dimensional data sets, it is computationally simple and fast, and the search in the feature space

is separated from the search in the hypothesis space. The next sections describe the feature extraction and selection methods.

4.3.1 Feature extraction

We obtain feature sets for coreference candidates, in which mentions are either a proper noun and a common noun (NAM-NOM), or both are common nouns (NOM-NOM). We denote a coreference candidate pair by (m_1, m_2). In the case of a NAM-NOM pair m_1 refers to the proper noun mention and m_2 to the common noun mention. As regards NOM-NOM, we consider two (m_1, m_2), pairs which differ by the order of the mentions, e.g. for the coreference candidate ("state", "country") we consider (m_1="state", m_2 = "country") and (m_1 = "country", m_2 = "state").

An (m_1, m_2) pair is processed as follows. We extract all RDF triples referring to m_1 from a knowledge source. In average we obtain 200 triples per mention. An RDF triple consists of subject, predicate and object. If m_1 is an object of a triple, we check if there is a partial string match between m_2 and the URI of the subject. In the other case, we check whether there is a string match between m_2 and the URI of the object. If the string match is observed, then we say that for a given coreference candidate pair we observe a feature named as the predicate of the RDF triple, and the feature is included into the feature set. If for RDF triples with a given predicate the string match never occurs in the entire training set, then the corresponding feature is not included into the feature set.

Examples of features for some of the mention pairs are presented in Table 4.1. Each mention is composed of the number of a document, the position in the document and the mention string itself. We select distinct sets of features for NAM-NOM and NOM-NOM mentions of two types of entities, namely person (PER) and geopolitical entities (GPE).[6] Consequently from

[6]Here we assume that the mentions in the corpus being processed are already annotated with their

Mention pair	**Feature**
1-225-Clinton, 1-87-president	http://www.w3.org/2004/02/skos/core#subject
529-324-Yasser Arafat, 529-402-leader	http://www.w3.org/2004/02/skos/core#subject
410-23-state, 410-109-country	http://www.w3.org/2004/02/skos/core#subject
2-637-Kuwait, 2-956-city	http://rdf.freebase.com/ns/location.country.capital
3-10-U.S.,3-892-States	http://www.w3.org/2002/07/owl#sameAs

Table 4.1: Feature examples

n_{1f}	number of instances in class 1 with feature f
$n_{1\bar{f}}$	number of instances in class 1 without feature f
n_{0f}	number of instances in class 0 with feature f
$n_{0\bar{f}}$	number of instances in class 0 without feature f
n_1	total number of instances in class 1
n_0	total number of instances in class 0
n_f	total number of instances with feature f
$n_{\bar{f}}$	total number of instances without feature f
n	total number of instances

Table 4.2: Feature examples

each of three background knowledge sources we extract four sets of binary features, namely NAM-NOM-GPE, NOM-NOM-GPE, NAM-NOM-PER, and NOM-NOM-PER. They typically contain 10-50 features. We apply the feature selection technique to each set.

4.3.2 Feature selection

In machine learning coreference candidates are called instances. We say than an instance belongs to class 1 if the mentions in the candidate pair are coreferent; 0 otherwise. Table 4.2 introduces some notation.

The chi-square feature selection metric, $\chi^2(f, c)$, measures the dependence between feature f and class $c \in \{0, 1\}$. If f and c are independent,

entity types.

then $\chi^2(f, c)$ is equal to zero. To select a class-relevant set of features, we utilized the following metric

$$\chi^2(f, c) = \frac{n(n_{1f}n_{0\bar{f}} - n_{0f}n_{1\bar{f}})^2}{n_1 n_f n_0 n_{\bar{f}}},$$

by averaging over the classes we obtain the metric for selecting a subset of features

$$\chi^2(f) = \sum_{i=0}^{1} Pr(c_i)\chi^2(f, c).$$

For example, we extract from Freebase a set of 22 features for the NAM-NOM pairs of mentions which refer to a GPE entity. After feature selection, the scores of 9 features are near to zero, consequently only 13 features should be considered. The two top-scoring features in this case are `http://www.w3.org/2002/07/owl#sameAs` and `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`. These features and their equivalents in other knowledge sources turned out to be highly relevant for other kinds of coreference as well.

4.4 Experiments

In this section we give some hints on the implementation of the model we used as a baseline, explain how the background knowledge is plugged into the model, and present the results of the experiments.

4.4.1 Baseline model definition

Tool selection

A recently introduced family of approaches to the task of coreference resolution try to represent the coreference task into some logical theory that supports the representation of uncertain knowledge. Among these approaches we can find a number of works [Poon and Domingos, 2008, Huang

et al., 2009, Culotta et al., 2007] based on the formalism called Markov logic [Domingos et al., 2008], which is a first-order probabilistic language which combines first-order logic with probabilistic graphical models.

In essence, Markov logic model is a set of first-order rules with weights associated to each rule. Weights can be learned from the available evidence (training data) or otherwise defined, and then inference is performed on a new (test) data. Such a representation of the model is intuitive and allows for the background knowledge be integrated naturally into it. It has been shown that the Markov logic framework is competitive in solving NLP tasks (see, for instance, [Poon and Domingos, 2007, Riedel and Meza-Ruiz, 2008], and *Alchemy* system documentation[7] for more references). Another advantage of the weighted first-order representation is that the model can be easily extended with extra knowledge by simply adding logical axioms, thus minimizing the engineering effort and making the knowledge enrichment step more straightforward and intuitive.

Given the above, the inference tool we have selected to be used in the coreference resolution tasks is the inference module of the Alchemy system, with Markov logic as a representation language.

The Alchemy inference module takes as inputs (i) a Markov logic model, that is, a list of weighted first-order rules, and (ii) an evidence database, that is, the list of known properties (true of false values of predicates) of domain objects. In the case of coreference resolution, domain objects are the entity mentions, and the properties they might have are gender, number, distance, semantic class, etc. In the following we discuss how these two parts of input are constructed.

[7] `http://alchemy.cs.washington.edu/`

Markov Logic Model

In defining a model for coreference resolution, we were inspired by Soon et al baseline [Soon et al., 2001], which uses the following features: pairwise distance (in terms of number of sentences), string match, alias, number, gender and semantic class agreement, pronoun, definite/demonstrative noun phrase and both proper names feature. This approach achieves an F-measure of 62.2% in the MUC-6 coreference task and of 60.4% on the MUC-7 coreference task.

A Markov logic model consists of a list of predicates and a set of weighted first-order formulae. Some predicates in our model correspond to Soon et al features: binary predicates such as *distance* between two entity mentions (in terms of sentences) and string match, and unary predicates such as *proper name*, *semantic class*, number (*singular* or *plural*) and gender (*male*, *female* or *unknown*). Also, we use *string overlap* in addition to *string match* and define yet another predicate to describe *distance*, which refers to the number of named entities of the same type between two given ones (e.g. if there are no other named entities classified as "person" between "Obama" and "President", the distance is 0). The predicate *corefer(mention,mention)* describes the relation of interest, and is called *query* predicate in Alchemy terminology, that is, we are interested in evaluating the probability of each grounding of this predicate given the known properties of all the mentions.

The second part of the model definition concerns constructing the first-order rules appropriate for a given task. We have defined the rules that connect the above properties of the mentions with the coreference property. Some of the examples are given below[8].

String match is very likely to indicate coreference for proper names,

[8]Full model is available at `https://copilosk.fbk.eu/images/1/1f/Coreference2.txt`

while for common nouns it is still likely but makes more sense in combination with a distance property:

$$20\ match(x, y) \wedge proper(x) \wedge proper(y) \rightarrow corefer(x, y)$$

$$3\ match(x, y) \wedge noun(x) \wedge noun(y) \wedge dist0(x, y) \rightarrow corefer(x, y)$$

The number before a formula corresponds to the *weight* assigned to it.

Gender and number agreement between two neighboring mentions of the same type provides a relatively strong evidence for coreference:

$$4\ male(x) \wedge male(y) \wedge singular(x) \wedge singular(y) \wedge follow(x, y) \rightarrow corefer(x, y)$$

We also define hard constraints, that is, crisp first-order formulae that should hold in any given world. Fullstop after the formula refers to an infinite weight, which, in turn, means that the formula holds with the probability equal to 1.

$$\neg corefer(x, x).$$

$$corefer(x, y) \wedge \rightarrow corefer(y, x).$$

We do not consider weight learning, so weights are assigned manually by tuning on a development set. We do not consider pronoun mentions as the background knowledge is relevant for proper name/common noun pairs in the first place.

Evidence database

The second input to the Alchemy inference module is an evidence database, i.e. the known values of non-query predicates listed in the previous section. Normally, the coreference resolution task is performed on a document corpus, in which each document is firstly preprocessed. Preprocessing consists in identifying the named entities (persons, locations, organization, etc.), as

well as their syntactic properties, such as part of speech, number, gender, pairwise distance, etc.

The data corpus we use for the experiments is ACE 2005 data set, with around 600 documents from the news domain. We work on a corpus in which each word is annotated with around 40 features (token and document ID, Part of Speech tags by TextPro[9], etc.). This allowed us to extract the syntactic properties of the mentions presented before. Note that for the gender property, we used male/female name lists to annotate proper names in the corpus. For common nouns, we defined two lists of gender tokens (which included "man","girl", "wife", "Mr.", etc.). The extracted properties are represented as evidences in the evidence database. Some examples of the properties (or evidences), we obtained are given below.

semclass ("2-83-Bob Dornan", "person")
neihgbourNouns ("2-82-Congressman","2-83-Bob Dornan")
propername ("2-83-Bob Dornan")
male ("2-83-Bob Dornan")
singular ("2-83-Bob Dornan")
pmatch ("2-740-Bob", "2-83-Bob Dornan")
match ("2-83-Bob Dornan", "2-942-Bob Dornan")
DBPedia_NAM-NOM_PER_2_type ("2-83-Bob Dornan", "2-62-Congressman")
YAGO_NAM-NOM_PER_1_type ("2-83-Bob Dornan", "2-86-Republican")

Inference

We worked on the gold standard annotation for named entities, and considered five named entity types: PERson, LOCation, GeoPoliticalEntity, FACility and ORGanization (although only PER and GPE were used in

[9]TextPro – `http://textpro.fbk.eu/`

the experiments presented later in this section). We worked on named and nominal entity mentions only. Alchemy inference was performed separately for each named entity type. Note that the size of the document corpus does not impact the quality of the results as documents are processed independently, one by one.

The Alchemy inference module, which takes as input the weighted Markov logic model and the database containing the properties of mentions, produces as a result the probabilities of coreference for each of NxN possible pairs of mentions, where N is the number of mentions:

$$coref(m_i, m_j) \quad p_{ij}, \quad 0 \leq p_{ij} \leq 1, \; i, j = \overline{1, N}$$

After having obtained this, we setup a probability threshold (e.g. $p = 0.9$) and consider only those pairs for which $p_{ij} \geq p$. On these pairs, we perform a transitive closure. Then the MUC scores [Vilain et al., 1995] are calculated. The resulting output consists of the list of coreference chains for each of the processed documents, and the MUC measures of the efficiency, namely, recall, precision and their harmonic mean (F1).

4.4.2 Injecting background knowledge into coreference model

In the Markov logic model, in addition to the syntactic predicates and rules described above, a set of predicates and rules that deal with background knowledge were introduced. The predicates, or pairwise semantic properties of mentions, are the most relevant features selected according to the methodology described in Section 4.3 from the DBpedia, YAGO and Freebase knowledge sources. The list of the selected features is given in Table 4.3.

The baseline Markov logic model is extended with the rules relating these semantic predicates with the coreference property. The arguments of a semantic predicate should be of the same named entity type (person or

KB name	NE type	Pair type	Property name
Freebase	GPE	NAM-NOM	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
Freebase	GPE	NAM-NOM	http://www.w3.org/2002/07/owl#sameAs
Freebase	PER	NAM-NOM	http://www.w3.org/2002/07/owl#sameAs
Freebase	PER	NAM-NOM	http://rdf.freebase.com/ns/people.person.profession
Freebase	PER	NOM-NOM	http://www.w3.org/2002/07/owl#sameAs
YAGO	GPE	NAM-NOM	type
YAGO	GPE	NAM-NAM	means
YAGO	PER	NAM-NOM	type
DBPedia	GPE	NAM-NOM	http://dbpedia.org/property/reference
DBPedia	GPE	NAM-NOM	http://www.w3.org/2004/02/skos/core#subject
DBPedia	GPE	NAM-NOM	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
DBPedia	PER	NAM-NOM	http://www.w3.org/2004/02/skos/core#subject
DBPedia	PER	NAM-NOM	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
DBPedia	PER	NAM-NOM	http://dbpedia.org/property/title

Table 4.3: Selected features

geopolitical entity), and the distance relation relation must hold between them. An example of a rule incorporating a semantic predicate is given below:

$$2.5\ YAGO\_NAM-NOM\_GPE\_1\_type(x,y) \wedge propername(x) \wedge noun(y) \wedge neihgbourNouns(x,y) \rightarrow corefer(x,y)$$

This rule states that if we extract the `rdf:type` feature (see Section 4.3.1) for a pair of GPE entity mentions ($YAGO\_NAM-NOM\_GPE\_1\_type(x,y)$), one entity mention is a proper name ($propername(x)$), another is a common noun ($noun(y)$), and there are no other non-pronominal entity mentions of the same type, i.e. GPE, between them in the document ($neighbourNouns(x,y)$), then they are likely to corefer.

For the experiments, the ACE data set was first ordered by the number of named entities linked to Wikipedia and split into two subsets of equal

size (*ACE-SUBSET-1* and *ACE-SUBSET-2*): odd documents from the ordered list formed the first subset, even formed the second one. *ACE-SUBSET-1* was used for feature selection and rule weights tuning , while on *ACE-SUBSET-2* the Markov logic model extended with background knowledge was tested. For the latter experiments, we have created yet another document set, *ACE-SUBSET-3*, which contains 50 documents from *ACE-SUBSET-2* with the highest background knowledge coverage (i.e. with the highest number of entity mentions linked to Wikipedia).

Tables 4.4 and 4.5 present MUC scores of the experiments for *ACE-SUBSET-2* and *ACE-SUBSET-3*, accordingly. Each table reports the values of MUC recall, precision and F1 for the models without and with the use of background knowledge extracted from DBpedia, YAGO and Freebase. Experiments were conducted for geopolitical entities (GPE) and persons (PER). Compared to the other three NE types (locations, organizations and facilities), persons and geopolitical entities constitute the major part of the corpus, so we do not report these results here. Additionally, improvement obtained when using background knowledge from LOD sources for the locations, organizations and facilities is insignificant. Also, we do not report the experiments for geopolitical entities with knowledge obtained from Freebase and DBpedia as the corresponding improvement for these cases was insignificant as well.

The improvement in $F1$ is 5% for GPE due to the use of YAGO on both datasets. The improvement in $F1$ for PER with the use of YAGO and Freebase is a bit higher for *ACE-SUBSET-3* (1.5% versus 2%) due to the increase of coverage in the latter. Relatively lower improvement for DBpedia as compared to YAGO and Freebase is most probably due to the fact that this knowledge source is much less structured and polished with respect to YAGO and Freebase.

NE type	KB	R	P	F1
GPE	none	0.7446	0.9371	0.8298
GPE	YAGO	0.8314	0.9308	**0.8783**
PER	none	0.7003	0.7302	0.7149
PER	DBpedia	0.7125	0.7196	0.7160
PER	Freebase	0.7178	0.7343	**0.7259**
PER	YAGO	0.7208	0.7348	**0.7277**

Table 4.4: MUC scores for GPE and PER NE types, *ACE-SUBSET-2* document set

NE type	KB	R	P	F1
GPE	none	0.7763	0.9380	0.8495
GPE	YAGO	0.8536	0.9335	**0.8918**
PER	none	0.7447	0.6946	0.7188
PER	DBpedia	0.7669	0.6852	**0.7238**
PER	Freebase	0.7749	0.7024	**0.7369**
PER	YAGO	0.7785	0.7039	**0.7393**

Table 4.5: MUC scores for GPE and PER NE types, *ACE-SUBSET-3* document set

4.5 Conclusion and future work

In this chapter we have applied our methodology for supporting a natural language processing task with semantic information available in LOD to the task of intra-document coreference resolution. More specifically, we map the terms in the text to concepts in Wikipedia and then to LOD resources linked to Wikipedia (DBpedia, Freebase and YAGO). We have proposed a method for selecting a subset of knowledge relevant for solving the coreference task which is based on feature selection algorithms. Automatic feature selection is an important point of our approach. Note that we make no prior assumptions on the structure of the LOD knowledge sources.

We have implemented the coreference resolution process with the help of the inference module of the Alchemy tool. The latter is based on Markov logic formalism and allows combining logical and statistical representation and inference. We have conducted evaluation on the ACE 2005 data set. The results show that usage of semantic knowledge results in increase the overall MUC F_1 measure, due to the increase in recall

Future work directions include further exploiting the Linked Data resources (including the one not used in this chapter, e.g. Cyc)to extract more properties and rules to support coreference resolution, as well as using the links between different Linked Data resources to obtain more knowledge. Also, we are interested in experimenting with the full task, which includes named entity recognition module and learning the weights of the formulae of the model from the training data.

Chapter 5

Semantic relation extraction between pairs of nominals

In this chapter we apply our framework to the task of semantic relation extraction between pairs of nominals. We show that usage of Wikipedia as semantic mediator in this case has certain limitations, and analyze the reasons. Nevertheless, we show that even without employment of disambiguation techniques semantic relation extraction between nominals can be improved by combining semantic features with shallow syntactic processing. We obtain semantic feaures from WordNet, OpenCyc and YAGO, and define kernels measuring the similarity of pairs of nominals in the context in terms of shallow syntactic features and generalizations of the nominals. In this chapter we describe an extension of our original approach ranked 2nd in the Task 8, "Semantic Relation task between nominals", during the SemEval 2010 evaluation campaign. The extension outperforms the original approach.

5.1 Introduction

In this chapter we describe the application of our methodology to semantic relation extraction (SRE) between nominals. More specifically, we have conducted the SRE case study on the SemEval-2010 Task 8 "Multi-Way Classification of Semantic Relations Between Pairs of Nominals." The task consists in identifying which semantic relation holds between two nominals in a sentence [Hendrickx et al., 2010]. The set of relations is composed of nine mutually exclusive semantic relations and the *Other* relation. The task requires to return the most informative relation between the specified pair of nominals, e_1 and e_2, in a context taking into account their order.

Our study is motivated by the Task 8 annotation guidelines which suggest that semantic knowledge about e_1 and e_2 plays a very important role in distinguishing among different relations. For example, relations *Cause-Effect* and *Product-Producer* are closely related, and one of the restrictions which might help to distinguish between them is that products must be concrete physical entities, while effects must not. This motivated us to focus on semantic features obtained from various sources of background knowledge (BK), e.g. ResearchCyc, OpenCyc, WordNet, DBpedia, and YAGO.

In this chapter we present an extension of our previous work [Tymoshenko and Giuliano, 2010] on using ResearchCyc as a source of semantic knowledge for SRE in SemEval-2010 evaluation campaign. The work was based on the approach by Giuliano et al. [2007a] implemented as *JSRE*[1] tool. Both current and previous approaches exploit two information sources: (i) the contextual and syntactic information from the sentence where the nominals appear, and (ii) semantic information. In [Giuliano et al., 2007a] the latter was represented by WordNet synonymy and hyperonymy informa-

[1] `http://hlt.fbk.eu/en/technology/jSRE`

tion, while in [Tymoshenko and Giuliano, 2010] we employed information of similar nature from ResearchCyc. The different kinds of information were represented by different kernel functions. We used support vector machines [Vapnik, 1998] as a classifier. The [Tymoshenko and Giuliano, 2010] version of the system achieved an overall F_1 of 77.62%, scoring second in the final ranking.[2]

In the current work we extend the semantic kernel family of *JSRE* with kernels based on information from OpenCyc, WordNet,[3] DBpedia, and YAGO, pursuing two objectives. First, we investigate whether the idea of our framework to use Wikipedia as a semantic mediator for disambiguation is applicable to nominals. We compare it to the baseline disambiguation strategies: using most frequent sense and using all senses of a given nominal. Second, we investigate which knowledge source/knowledge source combination(s) is(are) most beneficial for the SRE task.

Regarding the first objective, based on the results of the experiments we conclude that the *all-senses* strategy is preferred both to the *most frequent sense* strategy, and, at the current level of development of Linked Open Data (LOD) (see Chapter 2), to our framework as well. We provide an analysis of the problems encountered by our framework when disambiguating semantic nominals in terms of LOD URIs. The analysis shows that most of the problems originate from absence of Wikipedia pages corresponding to the very high-level abstract generic concepts and missing mappings between resources. As regards the second objective, we conclude that WordNet and OpenCyc have high coverage for the general-domain semantic nominals, and they both give improvement as compared to purely syntactical features. Combination of semantic kernels based on informa-

[2] *FBKIRST-COMBO12VBCA* in `http://semeval2.fbk.eu/semeval2.php?location=Rankings/ranking_task8.html`

[3] Note that Giuliano et al. [2007a] were provided with the gold-standard mappings from nominals to WordNet synsets [Girju et al., 2007]. Unlike them we have to establish the mappings by ourselves.

tion extracted from WordNet and OpenCyc without any disambiguation combined with purely syntactic shallow linguistic (SL) kernel by Giuliano et al. [2006], results in F_1 measure of 81.8% on SemEval test data, outperforming both our previous system [Tymoshenko and Giuliano, 2010] and SL kernel alone by 4%.

This chapter is structured as follows. First we overview the usage of semantic knowledge in the task of relation extraction in general domain in Section 5.2. Then, we present a kernel-based approach to SRE and a family of semantic bag-of-generalizations kernels in Section 5.3. In Section 5.4.2 we report the performance of disambiguation strategies, provide analysis of coverage of BK sources, and provide error analysis of our framework. Finally, in Section 5.4.3 we report the results of SRE experiments and discuss them.

5.2 Related work

Supervised relation extraction (RE) can be cast as feature vector classification [Tratz and Hovy, 2010, Zhou et al., 2005], Bayesian network inference [Roth and Yih, 2002], or kernel-based classification where kernels may be linear [Giuliano et al., 2006] or operate upon more complex structures, such as parse trees [Zelenko et al., 2003, Bunescu and Mooney, 2005, Culotta and Sorensen, 2004, Nguyen et al., 2009]. Consistently with the other NLP tasks, performance of different machine learning algorithms, e.g. the algorithms listed above, depends on the features they employ. In this section we describe the semantic feature subset frequently employed in RE for the pairs of nominals or named entity mentions.

Coarse-grained named entity types. Vast majority of approaches to RE exploit *coarse-grained entity types*, such as *person* or *location*. They can be either used as input data for a RE algorithm [Nguyen

et al., 2009, Bunescu and Mooney, 2006, 2005, Zhou et al., 2005, Giuliano et al., 2007b], or can be jointly inferred along with the relation labels [Roth and Yih, 2002]. The labels can be produced by a Named Entity Recognition (NER) tool or mined from thesauri or gazetteers, for example, U.S. Census Gazetteers and Roget's thesaurus divisions [Tratz and Hovy, 2010].

Semantic classes and relations. *Semantic classes* of nominals or entity mentions of interest or *semantic relations* between them[4] are another popular kind of features. So far, WordNet semantic network has been one of the most popular sources of such features. Majority of approaches in the general domain exploit WordNet alone or combine it with the other sources of background knowledge. For example, WordNet is used to obtain semantic classes of nominal of interest, define whether nominals or entity mentions of interest are in relations of hypernymy/hyperonymy or holonymy/meronymy, words in the synset glosses [Zhou et al., 2005, Chen et al., 2010, Tratz and Hovy, 2010, Giuliano et al., 2007a, Negri and Kouylekov, 2010, Rink and Harabagiu, 2010, Hendrickx et al., 2007]. In the recent years Wikipedia has also been used for this purpose. For example, Wikipedia has been used to extract a feature indicating whether entity mentions are in *parent-child* relationship [Chan and Roth, 2010].

Semantic relatedness. Another set of features are those indicating *semantic relatedness of entity mentions or nominals* of interest. In order to extract such features Chan and Roth [2010] exploited Wikipedia, while Szarvas and Gurevych [2010] exploited co-occurrence statistics and semantic relatedness measure based on usage of WordNet, Wiktionary[5] and Wikipedia-based Explicit Semantic Analysis [Gabrilovich

[4]naturally other relations than those of interest

[5]`http://www.wiktionary.org/`

and Markovitch, 2007].

Cooccurrence information. A set of approaches uses corpora to obtain *clusters of words frequently co-occurring with entity/concept mentions* of interest and uses them as features [Chan and Roth, 2010, Hendrickx et al., 2007, Rink and Harabagiu, 2010].

Note that the majority of the approaches does not limit their semantic feature sets only to one feature kind. For example, the authors of the top-performing SemEval-2010 system in Task 8, the task that we are investigating in this chapter, Rink and Harabagiu [2010] used semantic features indicating semantic properties of distinct nominals and properties of pairs of nominals, among others. The former included WordNet hypernyms, VerbNet [Schuler, 2005] verb classes, clusters of words related to each nominal of interest extracted from Google N-Gram data.[6] The latter included patterns returned by the TextRunner tool [Yates et al., 2007]. The patterns consist in most common phrases occurring in between nominals of interest in both directions in external corpora. Additionally, Rink and Harabagiu [2010] extracted features from FrameNet- [Fillmore et al., 2003] and PropBank-style[7] annotations. Their system scored first in SemEval 2010 Task 8 evaluation, achieving macro-average F_1 measure over all the relations of 82.19%. Note that even though in this chapter we employ a considerably simpler feature set, we achieve a comparable result of 81.8%.

Recently, emergence of large-scale knowledge bases such as YAGO and Freebase promoted a set of weakly supervised approaches, called *distant supervision* approaches, that use facts from the knowledge bases as the relation seeds. Since distant supervision employs the background knowledge not for the purpose of feature extraction, we do not describe such approaches in this section. However, their brief overview is available in

[6] http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2009T25

[7] http://verbs.colorado.edu/~mpalmer/projects/ace.html

Section 6.6.

5.3 Kernel methods for Relation Extraction

In order to implement the approach based on shallow syntactic and semantic information, we employed a linear combination of kernels using the support vector machines as a classifier. We use two types of basic kernels: syntactic and semantic kernels. They were combined by exploiting the closure properties of kernels. As in [Giuliano et al., 2006] we define the composite kernel $K_C(x_1, x_2)$ as follows.

$$\sum_{i=1}^{n} \frac{K_i(x_1, x_2)}{\sqrt{K_i(x_1, x_1)K_i(x_2, x_2)}}. \tag{5.1}$$

Here x_1 and x_2 are vectors and n is the total number of basic kernels. Each basic kernel K_i is normalized.

All the basic kernels are explicitly calculated as follows

$$K_i(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2) \rangle , \tag{5.2}$$

where $\varphi(\cdot)$ is the embedding vector. The resulting feature space has high dimensionality. However, Equation 5.2 can be efficiently computed explicitly because the representations of input are extremely sparse.

5.3.1 Shallow syntactic kernels

We employ shallow syntactic kernels taking into account local context of semantic nominals, their global context, and the combination of the two, i.e. the shallow linguistic (SL) kernel proposed by Giuliano et al. [2006].

Local context kernel

Local context is represented by terms, lemmata, PoS tags, and orthographic features extracted from a window around the nominals considering

the token order. Formally, given a relation example R, we represent a local context $LC = t_{-w}, ..., t_{-1}, t_0, t_{+1}, ..., t_{+w}$ as a row vector

$$\psi_{LC}(R) = (tf_1(LC), tf_2(LC), ..., tf_m(LC)\ ) \in \{0,1\}^m, \tag{5.3}$$

where tf_i is a feature function which returns 1 if the feature is active in the specified position of LC; 0 otherwise. The local context kernel $K_{LC}\,(R_1, R_2)$,[8] is defined as

$$K_{LC\_e1}(R_1, R_2) + K_{LC\_e2}(R_1, R_2), \tag{5.4}$$

where K_{LC_e1} and K_{LC_e2} are defined by substituting the embedding of the local contexts of e_1 and e_2 from R_1 and R_2 into Equation 5.2, respectively,.

Global context kernel

Giuliano et al. [2006] introduce three global context kernels following the hypothesis by Bunescu and Mooney [2006] which suggests that a sentence expresses the relation between two entity mentions e_1 and e_2 according to one of the following patterns:

- **Fore-Between (FB) context**. Words before e_1 and between e_1 and e_2.
- **Between (B) context**. Words between e_1 and e_2.
- **Between-After (BA) context**. Words between e_1 and e_2 and after e_2.

Giuliano et al. [2006] represent the above-listed patterns by means of bags-of-words populated with n-grams. Given a pattern P (where P is a FB, B or BA) and a relation example R, they represent R as a vector:

$$\psi_P(R) = (tf_1(P), tf_2(P), ..., tf_m(P)\ ) \in R^m, \tag{5.5}$$

[8] where R_1 and R_2 are the relation instances being compared

where $tf_1(P)$ indicates how many times a specific k-gram (with k taking different values) occurs in pattern P.

The *global context* kernel is defined as:

$$K_{GC}(R_1, R_2) = K_{FB}(R_1, R_2) + K_B(R_1, R_2) + K_{BA}(R_1, R_2), \tag{5.6}$$

where K_{FB}, K_B, K_{BA} are obtained by substituting the embeddings of R_1 and R_2 defined by Equation 5.5 into Equation 5.2.

Shallow linguistic kernel

Finally, the shallow linguistic (SL) kernel combines local and global information as follows:

$$K_{SL}(R_1, R_2) = K_{LC}(R_1, R_2) + K_{GC}(R_1, R_2) \tag{5.7}$$

5.3.2 Semantic kernels

The semantic kernels incorporate the semantic information as a bag of generalizations (BOG). Here, class a is a generalization of class b if all elements of b are elements of a. Class a is a generalization of an individual B, if a is a type of B or a generalization of type of B. All the semantic kernels follow the same pattern described below, with the only difference in the way the generalizations are obtained (see Section 5.3.3).

Formally, given a relation example R we represent the generalizations of a nominal e, BOG, as

$$\psi_{BOG}(e) = (fc(c_1, e), ..., fc(c_k, e)) \in \{0, 1\}^k, \tag{5.8}$$

where the binary function $fc(c_i, e)$ shows if a particular semantic class c_i is contained in BOG.

The *bag-of-generalizations* kernel $K_{genls}(R_1, R_2)$ is defined as

$$K_{genls\_e1}(R_1, R_2) + K_{genls\_e2}(R_1, R_2), \tag{5.9}$$

Symbol	Explanation
e	semantic nominal mention
e^l	lemma of e
e^{wiki}	link to Wikipedia assigned to e by TWM
$e^{mfs-wiki}$	most frequent sense of e in the TWM training set for e
e^{db}	DBpedia URI corresponding to e^{wiki}
e^{mfs-db}	DBpedia URI corresponding to $e^{mfs-wiki}$

Table 5.1: Notation

where K_{genls_e1} and K_{genls_e2} are defined by substituting the embedding of BOG (Equation 5.9) of e_1 and e_2 into Equation 5.2 respectively.

5.3.3 Semantic kernel instantiation

In this section we describe our methods to populate a bag of generalizations (BOG) for a nominal using either its lemma, or a link to DBpedia/Wikipedia. Table 5.1 explains the notation.

OpenCyc. We have devised three different versions of BOG based on information from OpenCyc, **OpenCycAll**, **OpenCycDis** and **OpenCycMfs**.

In order to instantiate *OpenCycAll*, we query OpenCyc for all triples matching the pattern *(?uri, ?p, "e_l")*, where *?uri* and *?p* are variables. We do not use a specific vocabulary term, e.g. `rdfs:label` instead of variable *?p*, in order to increase the recall of retrieved *?uri*-s. For each retrieved *?uri* we extract all its generalizations, defined by term `rdfs:subClassOf`, along with the transitively inferred generalizations, and add them to BOG.

We instantiate *OpenCycDis* using our framework. Specifically, we look for Cyc constants connected to e^{db} by means of `owl:sameAs` link, and obtain their generalizations as described above. *OpenCycMfs* is OpenCycDis, with e^{mfs-db} employed instead of e^{db}.

WordNet. We instantiate three WordNet-based *BOG*s, **WordNetAll**, **WordNetMfs** and **WordNetDis**.

WordNetAll is a *BOG* populated with the synset identifiers of all the hypernyms, direct and inherited,[9] of all the synsets containing e^l. *WordNetMfs* is populated with all the hypernyms of the most frequent sense of lemma of e according to the built-in WordNet sense frequency statistics.

WordNetDis is a *BOG* instantiated using our framework, i.e. by using a DBpedia URI produced by TWM as a mediator between e and a WordNet synset. Here we encounter the problem that there is no native DBpedia `owl:sameAs` mapping between DBpedia "things" and WordNet synsets. DBpedia employs the WordNet URIs[10] only to define the classes of the "things", based on manual mapping between Wikipedia infoboxes and WordNet synsets. Moreover, given that (1) Wikipedia pages corresponding to nominals typically do not have an infobox, and (2) manual mapping might not cover all the infoboxes, we need to look for alternative ways.

Therefore, we obtain the WordNet synset-DBpedia "thing" mapping from an external resource, called BabelNet [Navigli and Ponzetto, 2012]. It contains automatically produced mappings between Wikipedia pages (easily convertible to DBpedia URIs) and WordNet 3.0 synsets. Navigli and Ponzetto [2012] report that their mapping strategy achieves 78% F_1 measure. We use BabelNet to produce a set of `owl:sameAs` statements connecting the DBpedia URIs to the WordNet 3.0 VUA synset URIs (see Section 2.5.5). As a side effect we provide further manual evaluation of the BabelNet mappings.

[9] inherited hypernyms are all the hypernyms in the WordNet hypernym hierarchy that are in hypernymy relation with a given synset

[10] Semantic Web version of WordNet 2.0 created by [Van Assem et al., 2006]

ComboWordNet. Since BabelNet is a resource produced automatically, it might miss mappings. Moreover, WordNet might lack coverage for some e^l, typically named entities or domain-specific terms. In order to overcome these problems, we have devised two combined techniques, **ComboWordNetMaxDis** and **ComboWordNetMaxCov**

For a given nominal **ComboWordNetMaxDis** *BOG* is extracted as follows:

1. Check if e^{wiki} is mapped in BabelNet to a WordNet synset. If yes, add all the hypernyms of the synset to *BOG*, and stop. Otherwise, follow to the next step.

2. Check if there is an `owl:sameAs` mapping between e^{wiki} and a YAGO (see Section 2.5.3) concept/entity. If yes, then add all the WordNet-derived superclasses of this concept/entity to *BOG*, and stop. Otherwise, follow to the next step.

3. Check if YAGO contains classes based on Wikipedia categories of e^{wiki}. If yes, then add all their WordNet-derived YAGO generalizations to *BOG*, and stop. Otherwise, follow to the next step.

4. Collect plural heads of categories of e^{wiki}, if available. Add all the WordNet hypernyms of their most frequent WordNet sense to *BOG* and stop. If no plural heads are available, follow to the next step.

5. Instantiate *BOG* using the *WordNetAll* strategy.

ComboWordNetMaxCov is intended to achieve maximal coverage. Given e, we first try to populate the *BOG* using **WordNetAll**. In case if the *BOG* is empty, it follows steps 2-4 of the **ComboWordNetMaxDis** strategy.

5.4 Experiments

5.4.1 Experimental setup

We train and test the models on official SemEval 2010 Task 8 training and test datasets, comprising 8000 and 2716 sentences respectively. We use the official SemEval scorer to evaluate the results.

Sentences have been tokenized, lemmatized and PoS-tagged with TextPro [Pianta et al., 2008].[11] All the experiments were performed using jSRE customized to embed our kernels.[12] jSRE uses the SVM package LIBSVM [C. and L., 2001]. The task is cast as multi-class classification problem with 19 classes (2 classes for each relation to encode the directionality and 1 class to encode *Other*). The multiple classification task is handled with All-vs-All technique. The SVM parameters have been set as follows. The cost-factor W_i for a given class i is set to be the ratio between the number of negative and positive examples. We set the regularization parameter C to $C_{def} = \frac{1}{\sum K(x,x)}$, where x are all examples from the training set. The default value is used for the other parameters.

In order to enrich text with background knowledge as described in Section 5.3.3 we use WordNet VUA 3.0,[13] OpenCyc 4.0,[14] DBpedia 3.8 and core YAGO2 (version of 2012/01/09). We used TWM to annotate common nominals e_1 and e_2 with links to Wikipedia, using the whole sentences where they occur as a disambiguation context.

[11] `http://textpro.fbk.eu/`

[12] jSRE is a Java tool for relation extraction available at `http://tcc.itc.it/research/textec/tools-resources/jsre.html`.

[13] `http://thedatahub.org/dataset/vu-wordnet`

[14] `http://sw.opencyc.org/downloads/opencyc_owl_downloads_v4/opencyc-latest.owl.gz`

5.4.2 BK enrichment evaluation and discussion

The accuracy and coverage achieved by various *BOG* population strategies influence the further performance of SRE systems employing *BOG* kernels. In this subsection we evaluate coverage of semantic nominals by all the strategies described in Section 5.3.3, and the accuracy of Wikipedia link-based disambiguation strategies.

BK coverage evaluation

Table 5.2 reports the percentage of nominals that we were able to map to a BK source using the method specified in the first column. The *Wikipedia* line refers to the percentage of nominals annotated with non-null Wikipedia links by the Wiki Machine, and the remainder of abbreviations is described in Section 5.3.3.

The table shows that Wikipedia has the highest coverage, 97.09%. Note that this high number is achieved considering a nominal covered if TWM has a training set for it. However, in some cases, this training set might not contain a correct sense, for example, due to absence of a Wikipedia page describing a specific sense of a nominal. Sources created by limited groups of experts, such as WordNet and OpenCyc (represented by WordNetAll, OpenCycAll), also have high coverage for non-domain specific semantic nominals.

Usage of mappings between DBpedia and external resources results in drop of coverage. OpenCycDis and OpenCycMfs strategies resulted in a considerable drop of coverage of 43%, as compared to OpenCycAll. Using BabelNet as a mediator between Wikipedia predictions also results in 20% drop of coverage as suggested by WordNetDis. Note that with WordNet-DisMfs the drop is larger. This happens because typically the most frequent sense for a given nominal in the TWM training data is a specific named

Knowledge source	mapped (%)
Wikipedia	97.09
OpenCycDis	39.27
OpenCycMfs	35.39
OpenCycAll	87.51
WordNetAll, WordNetMfs	94.2
WordNetDis	74.86
WordNetDisMfs	67.14
ComboWordNetMaxDis, ComboWordNetMaxCov	97.48

Table 5.2: Coverage

	P	R	F_1
TWM	76	86	81
TWM-MFS	61	69	65

Table 5.3: TWM performance (in %)

entity/individual that is absent in OpenCyc and WordNet. Combined strategies, ComboWordNetMaxDis and ComboWordNetMaxCov, result in maximal coverage of 97.48.

Evaluation of TWM-mediated mappings

We evaluated the quality of TWM annotations and further DBpedia-BK source mappings on a small gold standard of 50 SemEval training corpus sentences each containing two nominals. We manually annotated 100 nominals of interest from these sentences with links to appropriate Wikipedia pages. Table 5.3 reports the performance of TWM in the first line. The performance of the system which always predicts the most frequent sense from TWM training data is reported in the second line (TWM-MFS).

We have manually analyzed the mappings to WordNet and OpenCyc obtained by DBpedia mediation, to be further employed for BK enrichment

Class	WordNet (%)	OpenCyc (%)
Correct	53	29
No sense in Wikipedia	12	12
TWM mistake	9	9
TWM null output	2	2
TWM related	4	4
BK concept missing	11	18
BK mapping wrong	1	4
BK mapping missing	5	21
BK mapping technical error	3	1

Table 5.4: Results of manual analysis of mappings produced by the framework

in OpenCycDis and WordNetDis. We have classified our observations in the list below and report their corresponding percentages in Table 5.4

1. **Correct.** TWM output is correct, and mapping from DBpedia to the BK source (WordNet via BabelNet, or OpenCyc via `owl:sameAs` links) is correct.

2. **No sense in Wikipedia.** Wikipedia mapping is wrong, because there is no Wikipedia page corresponding to this specific sense of the nominal. In *"The system as described above has its greatest application in an arrayed* ***configuration***$_{E1}$ *of antenna* ***elements***$_{E2}$*"*, **E1** is used in its generic sense of an arrangement of elements. There is no corresponding page in Wikipedia.

3. **TWM mistake.** TWM output is wrong, even though a page describing the correct sense is present in Wikipedia.

4. **TWM related.** Wikipedia page is a concept closely related to the concept meant by the nominal, but belongs to a different synset. In this specific task, such mapping is wrong. *"The* ***singer***$_{E1}$*, who performed three of the nominated songs, also caused a* ***commotion***$_{E2}$ *on*

the red carpet." TWM maps **E1** to the Wikipedia page `Singing`,[15] this page is closely related to the concept of *singer*, however its generalizations from the BK sources will provide erroneous information that **E1** is a process, while it is a person.

5. **TWM null output.** TWM does not output any mapping, due to absence of a training set for a specific nominal.

6. **BK concept missing.** TWM output is correct, however the concept is not present in BK source. In *"The solute was placed inside a beaker and 5 mL of the* ***solvent***$_{E1}$ *was pipetted into a 25 mL glass* ***flask***$_{E2}$ *for each trial."*, **E2** is correctly mapped to the `Laboratory_flask` page. WordNet contains a synset for the general notion of the flask as a "bottle that has a narrow neck", but contains no knowledge about the laboratory flask. In the majority of cases such concepts are domain-specific, and BK sources contain their generalizations only.

7. **BK mapping wrong.** TWM output is correct, the concept is present in BK source, but the mapping from DBpedia to the source is wrong. For instance, **E1** from the sentence from the previous item is correctly mapped to the DBpedia resource `dbpedia:Solvent`, however in OpenCyc `dbpedia:Solvent` is connected by means of `owl:sameAs` link with the `opencyc:FinanciallySolvent`[16] concept, that is "The quality or state of being financially able to pay all legal debts."

8. **BK mapping missing.** TWM output is correct and the concept is present in BK source, however, no information about the mapping is available. In *"It was a friendly* ***call***$_{E1}$ *to remind them about the*

[15]Original Wikipedia URL can be retrieved by adding `http://en.wikipedia.org/wiki/` before the page name

[16]Here and further the original OpenCyc URI may be recovered by substituting `opencyc:` to `http://sw.opencyc.org/2012/05/10/concept/en/`

bill$_{E2}$ *and make sure they have a copy of the invoice."*, **E1** is correctly mapped to `Telephone_call`, and both WordNet and OpenCyc contain this concept, as `wn30:synset-call-noun-1`[17] and `opencyc:MakingAPhoneCall` correspondingly, but neither BabelNet nor OpenCyc contain a corresponding mapping.

9. **BK mapping technical error.** Technical issues due to the constant change of Wikipedia, e.g. BabelNet mapping is correct, but it points to a redirection Wikipedia page.

5.4.3 SRE experiments and discussion

In this section we report the SRE experiments results. We used 10-fold cross-validation on SemEval training set to select the best kernel combinations, and tested them on the official SemEval test set.

Table 5.5 reports results obtained on the test set, and Table 5.6 reports results obtained on the training set in 10-fold cross-validation. We observe that all *SL + BOG* kernel combinations result in a substantial increase of macro-average F_1 as compared to *SL* only. The best kernel combination both in cross-validation on training and on the test set is *SL + WordNetAll + OpenCycAll*.

The difference between *SL + WordNetAll + OpenCycAll* and the second top result (*SL + WordNetMfs*) is statistically significant with $p < 0.05$. We used the approximate randomization procedure [Noreen, 1989] to compute the significance test.

Results obtained with *SL + ComboWordNetMaxDis* both on training and test set are lower than that of the other *SL + BOG* kernels. This is probably due to the noise introduced by the Wikipedia-based disambiguation strategy, current problems of which are described in Section 5.4.2.

[17]Here and further the original WordNet 3.0 URI may be recovered by substituting `wn30:` to `http://purl.org/vocabularies/princeton/wn30/`

Kernels	P	R	F_1
SL	72.35	80.3	76.03
SL + WordNetAll	77.85	84.19	80.8†
SL + WordNetMfs	77.96	84.17	80.83
SL + OpenCycAll	77.33	84.33	80.56
SL + WordNetAll + OpenCycAll	78.82	85.22	**81.8**†
SL + ComboWordNetMaxDis	76.25	82.73	79.26
SL + ComboWordNetMaxCov	77.87	84.01	80.72
Top Semeval-2010 system [Rink and Harabagiu, 2010]	82.25	82.28	82.19
Our best Semeval-2010 result [Tymoshenko and Giuliano, 2010]	74.98	80.69	77.62

Table 5.5: Overall performance on the test set, macro-average over all relation excluding "other". † indicates significant differences ($p < 0.05$). Significance tests are computed using approximate randomization procedure.

SL + ComboWordNetMaxCov, intended to increase the coverage of *WordNetAll* strategy, did not result in a significant improvement over *SL + WordNetAll*. We assume that this is due to the fact that WordNetAll already has very high coverage, and the 3% increase of coverage by *ComboWordNetMaxCov* (see Table 5.2) is too small to influence the results of the SRE experiments.

Line *SL + ComboWordNetMaxDis*^{MFS} of Table 5.6 reports the results of the experiments when *ComboWordNetMaxDis* is instantiated using most frequent Wikipedia sense instead of the TWM prediction. It is significantly[18] outperformed by *SL + ComboWordNetMaxDis*, showing the importance of employing word sense disambiguation when mapping to Wikipedia.

In Table 5.7 we compare per-relation performance of the baseline kernel, *SL*, to that of the two top-performing kernels, *SL + WordNetAll* and *SL + OpenCycAll + WordNet*. The table shows that semantic knowledge from WordNet is most important for the *Component-Whole* (+8.86%), *Product-Producer* (+8.13%), *Instrument-Agency* (+6.68%) relations. Adding se-

[18] $p < 0.001$, approximate randomization procedure used to compute the significance test

Kernels	P	R	F_1
SL	70.05	77.28	73.27
SL + WordNetAll	77.54	83.39	80.28
SL + WordNetMfs	76.24	82.58	79.16
SL + OpenCycAll	75.34	82.43	78.61
SL + ComboWordNetMaxDis	74.69	81.05	77.61
SL + ComboWordNetMaxDis^{MFS}	73.22	79.83	76.24
SL + ComboWordNetMaxCov	77.45	83.08	80.1
SL + WordNetAll + OpenCycAll	77.82	83.99	**80.71**

Table 5.6: Performance in 10-fold cross-validation on the training set, macro-average over all relations excluding "other"

Relation	SL	SL + WordNetAll	SL + OpenCycAll + WordNetAll
Cause-Effect	88.07	89.06 (+0.99)	90.08 (+1.02)
Component-Whole	65.99	74.85 (+8.86)	77.20 (+2.35)
Content-Container	80.94	83.53 (+2.59)	83.65 (+0.12)
Entity-Destination	85.07	86.99 (+1.92)	86.71 (-0.28)
Entity-Origin	78.82	83.82 (+5)	84.35 (+0.53)
Instrument-Agency	63.19	69.87 (+6.68)	72.20 (+2.33)
Member-Collection	78.52	84.09 (+5.57)	84.56 (+0.47)
Message-Topic	75.19	78.44 (+3.25)	81.70 (+3.26)
Product-Producer	68.46	76.59 (+8.13)	75.76 (-0.83)
Other	29.39	41.27 (+11.88)	45.18 (+3.91)

Table 5.7: Per-relation performance on the test set in terms of F_1 measure. Value in parentheses in the *SL+WordNetAll* column corresponds to the relative improvement as compared to *SL*. Value in parentheses in the *SL+WordNetAll+OpenCycAll* column corresponds to the relative improvement as compared to *SL+WordNetAll*

mantic features from OpenCyc is most beneficial for the *Message-Topic* relation (+3.26), and helps to further increase the F_1 for the *Component-Whole* (+2.35%) and *Instrument-Agency* relations (+2.33%).

5.5 Conclusion

In this chapter, we have reported the case-study in semantic relation extraction between pairs of common nominals. We have enriched the state-of-the-art kernel-based relation extraction system using shallow syntactic information, with new semantic kernels. All of them are simple bag-of-generalizations kernels, that differ by the source of generalizations and strategy employed to deal with ambiguity. We used OpenCyc, WordNet, YAGO and DBpedia as sources of semantic information. We tackled the ambiguity by using The Wiki Machine (TWM) (See Section 3.3) and using baseline techniques such as most frequent sense strategy or usage of all senses.

We have observed that when terms of interest to be enriched with semantic information are common nominals, our framework encounters problems. We have analyzed and classified the reasons of the problems. They include, for example, absence of Wikipedia pages corresponding to very general common-sense concepts, missing `owl:sameAs` mappings between the resources, and a number of other reasons that we have analyzed in detail in Section 5.3.

We have shown that external knowledge about generalizations of common nominals, encoded as a bag-of-generalizations kernel without any word sense disambiguation, significantly contributes to the improvement of the overall performance of the system. More specifically, we have demonstrated that the combination of semantic kernels based on information extracted from WordNet and OpenCyc without any disambiguation, combined with

purely syntactic shallow linguistic (SL) kernel by Giuliano et al. [2006], results in F_1 measure of 81.8% on SemEval test data, outperforming both our previous system [Tymoshenko and Giuliano, 2010] and SL kernel alone by 4%.

In future we plan to employ one of the off-the-shelf word sense disambiguation systems predicting WordNet senses in order to compare the impact of traditional disambiguation techniques to those of the baseline techniques. Additionally, it would be interesting to conduct an investigation, similar to the one presented in this chapter, on a corpus where terms of interest are named entity mentions or domain-specific terms. Our hypothesis would be that, in this case, Wikipedia mediation and knowledge resources other than WordNet (e.g. YAGO) would be of greater use.

Chapter 6

Biomedical entity relation mining

In this chapter we explore the use of semantic information from background knowledge sources for the task of relation mining between medical entities such as diseases, drugs, and their functional effects/actions. When conducting this research we have discovered that the biomedical resources currently available on LOD have limited coverage for the medical entities of interest, due to the proprietary nature of the data in the domain.

Therefore, we deviate from the first two steps of the framework, and employ alternative ways of extracting knowledge. We extract features from Wikipedia and specialized biomedical resources, including UMLS Semantic Network, MEDCIN, MeSH and SNOMED CT. Given that the resources might have different coverage, we propose a two-step approach. First, we learn multiple classifiers combining features from different resources, and correspondingly having different amount of semantic knowledge/coverage balance. Then we combine the predictions of the individual classifiers by means of an ensemble classifier. We show than in contrast to the general domain, semantic features can be highly discriminative, even in absence of syntactic evidences.

The author conducted this study while she was at Siemens Corporate Research, Princeton, NJ, U. S., for an internship under supervision of Dr. Swapna Somasundaran.

6.1 Introduction

Relation mining in the biomedical domain attempts to find interactions between medical entities. This can enable Clinical Decision Support (CDS) systems in performing critical functions such as identifying potentially adverse drug interactions from patient health records. Adverse drug interactions may occur due to a wide variety of factors involving ingredients of the drugs, their mechanisms of action within the body, their physiological effects, contraindications with certain conditions, etc. It is therefore important to build relation mining systems that can recognize such interactions with good accuracy.

State of the art approaches to relation mining (e.g. Frunza and Inkpen [2010], Rosario and Hearst [2004]) rely on human annotated corpora, where sentences containing entities of interest are annotated with their relation. This approach, however, is not feasible for our task due to the lack of human annotated corpora for all our clinical relations of interest.

In order to overcome this challenge, in this work, we exploit the hypotheses that *biomedical entities have certain inherent properties that are indicative of their interactions*, and *the way knowledge sources organize information regarding medical entities can be harnessed to infer their interactions*. Consequently, we exploit two different types of *entity-level semantics*. The first set of semantics correspond to the first hypothesis and is based on individual entity properties. For example, Aspirin, a drug, has a property of being anti-inflammatory, and anti-inflammatory drugs

have the property of treating pain. Thus, by using this knowledge and the knowledge that Headache is a type of pain, we can infer that the entity Aspirin is likely to have a *treat* relation with the entity Headache. The second set of semantics, corresponding to the second hypothesis, is based on the entity pair under consideration, and captures how information in standard knowledge sources links a given pair of entities. For example, a Wikipedia page for a drug typically mentions the diseases (or types of diseases) the drug treats in a "uses" subsection.

We test our hypotheses on the recognition of 10 different clinical relations from the National Drug File – Reference Terminology (NDF-RT)[1] using a number of knowledge sources such as the Wikipedia encyclopedia, Unified Medical Language System (UMLS) metathesaurus, that is a compilation of multiple biomedical vocabularies, and UMLS semantic network. We encode semantic features such as entity-category/taxonomy (derived from UMLS etc.) and entity-pair linkage information (derived from Wikipedia) into a machine learning algorithm. Based on the coverage and specificity of the resources and the features, we explore different feature combinations and construct different classifiers. Finally, we combine all the individual predictions using an ensemble approach.

Our investigations with entity-level semantic classifiers built using different knowledge source combinations reveal their strengths and weaknesses for large-scale biomedical relation mining. We compare our approach to distant supervision-based approaches that have been shown promising for relation mining between named entities (e.g. Mintz et al. [2009]). Experiments carried out over 97,000 entity pairs reveal that in the biomedical domain, distant supervision-based approaches that use sentence-level information face a number of challenges in terms of coverage and performance. Our approach that employs entity-level semantics from various knowledge

[1] `http://evs.nci.nih.gov/ftp1/NDF-RT/`

sources is able to achieve substantial improvements in both: we get an average improvement of 44 percentage points in coverage and 39 percentage points in performance (F_1). Finally, we show that even a simple ensemble approach that combines all the semantic information is able to get the best coverage and performance.

6.2 Entity-level semantics

Relation mining approaches for named entities such as Persons and Organizations have exploited human annotated corpora, such as ACE [ACE, 2000-2005], to construct systems that leverage linguistic and contextual information within text surrounding a given pair of co-occurring entities. This approach is not feasible for our task due to the absence of an annotated corpus for our relations of interest. However, to our advantage, biomedical relations are characterized by the properties of the involved entities. Additionally, the clinical domain has a number of knowledge sources providing information about medical entities in an organized fashion. We call this entity-level semantics, and harness it to develop our relation mining system.

Our relation mining is motivated by the goal to assist Clinical Decision Support (CDS) Systems in identifying and flagging adverse drug interactions. Specifically, we focus on drugs and a subset of their interactions with other medical entities in the NDF-RT ontology. The medical entities of interest in this work are: Drugs, Diseases, Drug Pharmacology (Chemical) Class, Drug Physiological Effects, Drug Ingredients, Drug Mechanism of Action. Table 6.1 describes the relations of interest involving these entities. Notice that drugs can have different types of relations with the same types of medical entities (e.g. Mechanism of Action).

As mentioned previously, entity-level semantics involves two different

Name	Description
may_treat	Drug A may treat Disease B
may_prevent	Drug A may prevent Disease B
may_diagnose	Drug A may diagnose Disease B
induces	Drug A induces Disease B
CI_with	Drug A is contraindicated (known to cause adverse reaction) with Disease B
has_Ingredient	Drug A has Ingredient B
has_PE	Drug A has Physiological Effect B
has_MoA	Drug A has Mechanism of Action B
CI_MoA	Drug A is contraindicated with Mechanism of Action of drug B
CI_ChemClass	Drug A is contraindicated with Chemical Class of drug B

Table 6.1: Relations of interest from NDF-RT

types of information. The first, *entity-specific semantics*, is based on the individual entity's properties and the second, *entity pair linkage*, is based on information on how the entities are linked in knowledge sources. For instance, the drug Aspirin is a type of analgesic (painkiller) drug that has the property of treating diseases (conditions) or symptoms involving pain, such as Headache and Toothache. This is an example of the first type of entity-level semantics where the class and taxonomic information of the drug and the disease clue their interaction. As an example for the second type of entity-level semantics, let us consider the Wikipedia page for the drug Ibuprofen. The page mentions the condition Fever under "Medical Uses". Similarly, Wikipedia pages for drugs Paracetamol and Codeine also have "Medical Uses" subsections where the symptoms that they cure are listed. Here, the manner in which a knowledge source such as Wikipedia links the two entities can clue to the type of relation between them.

We use Wikipedia[2], UMLS semantic network[3], and UMLS metathesaurus resources such as MEDCIN[4], SNOMED–CT[5] and MeSH[6] as our knowledge sources. All resources are used for extracting category and taxonomy information, while Wikipedia is used to capture linkage semantics.

6.3 Semantic features

Our semantic features can be broadly categorized as entity-specific features and entity pair features. The former includes category/taxonomy-based features while the latter includes link-based features for entity pairs.

6.3.1 Entity-specific features

These features are based on the category of the entity and capture the class properties of the individual entities. Categories and taxonomy represent topical and semantic class information about the entities. Category features are extracted from all knowledge sources listed above. Some of the entity specific features are as follows.

- **wikiCategory**. This is a set of features that capture the category of the Wikipedia page corresponding to an entity e, and its ancestors in the Wikipedia category taxonomy up to two levels up. For instance, the page for `Aspirin` has categories `Acetate_esters` and `Antiplatelet_drugs`.

[2] `http://www.wikipedia.org/`

[3] `http://semanticnetwork.nlm.nih.gov/`

[4] MEDCIN was created and is maintained by Medicomp Systems, Inc.(`http://www.medicomp.com/`). We have been using the version of MEDCIN available as a part of UMLS release (`http://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/MEDCIN/`)

[5] SNOMED CT is owned, maintained and distributed by the International Health Terminology Standard Development Organisation (IHTSDO). `http://www.ihtsdo.org/snomed-ct/`. We have been using the version of SNOMED CT available as a part of UMLS release (`http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html`)

[6] `http://www.ncbi.nlm.nih.gov/mesh`

- **umlsPF**. These features capture the taxonomical path information in various UMLS knowledge sources. Path is calculated from an entity of interest e to the root of a specific UMLS source and represented as: $[root.node_n.node_{n-1}.<...>.node_0]$, where $node_0$ is a direct parent of e in a source, and $node_{i+1}$ is a parent of $node_i$. *umlsPF* feature set also includes more generic subpaths of the full path shown above. For example, the following subpaths are also created as features: $[root]$, $[root.node_n]$, $[root.node_n.node_{(n-1)}]$, ... , $[root.node_n.node_{(n-1)}.....node_1]$. Depending on the knowledge sources, there are different feature sets:
  - **umlsPF:::SNOMED**. This is umlsPF with SNOMED CT as the source. For instance, for *Aspirin*, umlsPF:::SNOMED would include [*Drug or medicament.Musculoskeletal system agent. Anti-rheumatic agent. Anti-inflammatory agent. Non-steroidal anti-inflammatory agent .Salicylate*].
  - **umlsPF:::MSH**. This is umlsPF with MeSH as the source. For instance, for *Aspirin*, it would include [*Chemicals and Drugs (MeSH Category).Organic Chemicals.Phenols.Hydroxybenzoic Acids.Salicylic Acids*].
  - **umlsPF:::MEDCIN**. This is umlsPF with MEDCIN as the source. For instance, for *Aspirin*, it would include [*therapy.medications and vaccines.analgesics.salicylates*].
- **umlsSemType**. This feature set captures the semantic types of an entity in the UMLS Semantic Network, and is similar to wikiCategory features. For example, *Aspirin* has UMLS semantic types *Organic Chemical* and *Pharmacologic Substance*.
- **umlsCUI**. This is the UMLS Concept Unique Identifier (CUI) of an entity, (e.g. *C0004057* for *Aspirin*) and captures the identity of the entity.

6.3.2 Entity pair linkage features

The entity pair linkage features capture how information about one entity refers to the other entity, or how both entities refer to other concepts that are common to them. We encode two different types of entity features using Wikipedia as the knowledge source. In this work, we only focus on the linking and subsectioning information.

- **pairwiseLinkFeature**. These consider direct links between entities. There are two types of pairwise link features: (1) name of the subsection(s) in which the Wikipedia page corresponding to entity e_1 points to the Wikipedia page about entity e_2; (2) the same information in the opposite direction. For example, a link to the `Aspirin` page occurs in the *Prevention* subsection of the `Migraine` page, while the reverse link occurs in the *Medical uses* subsection of the `Aspirin` page.

- **sectLinkSectPath**. This feature set captures indirect links between the entities and includes the concatenated names of the subsections of Wikipedia pages corresponding to e_1 and e_2 having common outgoing links. For example, `Aspirin` links to `Tension_headache` in its *Medical uses* subsection, and `Migraine` links to `Tension_headache` in its *Cause* subsection. Thus the sectLinkSectPath path constructed for the `Aspirin` – `Migraine` entity pair is *Medical_Uses:::Cause*

6.4 Experiments

We perform experiments in two parts. In the first part (Section 6.4.3), we evaluate the utility of using entity-level semantics over a standard approach. The insights from the first part are then used to create an overall better relation recognizer in the second part (Section 6.4.4).

6.4.1 Data

We extracted the experimental dataset from the National Drug File– Reference Terminology (NDF-RT). NDF-RT is an extended formal ontological version of the National Drug File (NDF), a list of drugs and their properties released by U.S. Department of Veterans Affairs, Veterans Health Administration (VHA). It contains information about drugs and their relations with other biomedical entities, including interactions, physiological effects, methods of action, etc.

Entity pairs are extracted from the NDF-RT ontology, which provides the relation labels for each entity pair. Given an entity pair, we construct features based on entity-level semantics described above. This is then used to train a supervised relation classifier.

The dataset is a set of labeled examples. An example is a triple (e_1, R, e_2), where e_1 (subject) and e_2 (object) are UMLS entities corresponding to NDF-RT entities. R is either one of the NDF-RT relations listed in Table 6.1, or, if e_1 and e_2 are not related, $R = NOREL$ (and the entity pair is considered as a negative example).

We extracted positive examples by searching NDF-RT for all the entity pairs engaged in a given relation of interest. All entity pairs having more than one relation in NDF-RT were discarded to remove ambiguity during evaluation. Additionally, entities with symbols in their name (e.g. "%", ",", "/") were discarded, as these entities are likely to have no coverage in the knowledge sources (for our systems as well as the baseline). Negative examples were randomly generated following the closed world assumption. We randomly draw (e_1, e_2) and check whether NDF-RT contains information about relation between them. If it does not, then the entity pair is considered an example of a $NOREL$ relation.

The resulting dataset, $AUTONDF$, contains 48,519 positive and 48,519

negative examples. The number of entity pair examples per relation are as follows, CI_ChemClass: 1,113; CI_MoA: 318; CI_with: 13,819; has_Ingredient: 1,630; has_MoA: 6,509; has_PE: 10,449; induces: 271; may_diagnose: 386; may_prevent: 882; may_treat: 13,142; NOREL: 48,519.

For each (e_1, R, e_2) example we extract a set of features described in Section 6.3. In order to obtain features from UMLS Semantic Network and UMLS Metathesaurus features, we queried the off-line distribution of UMLS for CUIs of interest. Wikipedia-based features were extracted using JWPL Wikipedia API [Zesch et al., 2008], from the Wikipedia version of December, 2011[7]. If there was more that one page retrieved for either e_1 or e_2, all the pages were exploited as feature sources.

Due to size limitations of knowledge sources, they may not have coverage over all instances. For example, one or both entities in a pair may not have a corresponding page in Wikipedia, making it impossible to extract Wikipedia-based features. When training a classifier, instances that do not find coverage in the knowledge sources it uses are skipped.

6.4.2 Baseline

Our baseline, *DS*, is a system using distant supervision and sentence-level features. This approach has been suggested to circumvent the lack of sufficiently large, labeled corpus for relation extraction Mintz et al. [2009]. In distant supervision, for each pair of entities that are in a particular relation, all sentences containing those two entities are extracted from a large unlabeled corpus and a relation classifier is trained using textual features of these sentences. The underlying hypothesis is that "if entities e_1 and e_2 are known to be in relation R, then any sentence containing a mention of both e_1 and e_2 is likely to express the relation R".

We built *DS* using our *AUTONDF* dataset and PubMed as the source

[7] `http://dumps.wikimedia.org/enwiki/20111201/`

Relation	DS-covered			
	Count (Coverage)	P	R	F_1
CI_ChemClass	138 (12.40%)	61.90	9.42 (1.17)	16.35 (2.29)
CI_MoA	0 (0%)	0.00	0.00 (0.00)	0.00 (0.00)
CI_with	905 (6.55%)	83.63	20.88 (1.37)	33.42 (2.69)
has_Ingredient	64 (3.93%)	93.75	23.44 (0.92)	37.50 (1.82)
has_MoA	48 (0.74%)	77.78	29.17 (0.22)	42.42 (0.43)
has_PE	117 (1.12%)	0.00	0.00 (0.00)	0.00 (0.00)
induces	60 (22.14%)	0.00	0.00 (0.00)	0.00 (0.00)
may_diagnose	24 (6.22%)	0.00	0.00 (0.00)	0.00 (0.00)
may_prevent	183 (20.75%)	43.75	7.65 (1.59)	13.02 (3.06)
may_treat	2320 (17.65%)	59.22	98.66 (17.42)	74.02 (26.92)
NOREL	324 (0.67%)	66.67	0.62 (0.00)	1.22 (0.01)
Overall	4183 (4.31%)	44.25	17.26 (2.11)	19.81 (3.38)

Table 6.2: Baseline system performance.

Relation	Best Feature Set	Count (Coverage)	P	R	F_1
CI_ChemClass	umlsPF:::SNOMEDCT, umlsSemType, umlsCUI	939 (84.37%)	91.39	94.99 (80.14)	93.16 (85.4)
CI_MoA	wikiCategory, pairwiseLinkFeatures, umlsCUI	115 (36.16%)	95.65	95.65 (34.59)	95.65 (50.81)
CI_with	umlsPF:::SNOMEDCT, umlsCUI	10571 (76.50%)	93.41	94.76 (72.49)	94.08 (81.63)
has_Ingredient	umlsPF:::MEDCIN, wikiCategory, pairwiseLinkFeatures, sectLinkSectPath	134 (8.22%)	83.33	67.16 (5.52)	74.38 (10.36)
has_MoA	umlsSemType, umlsCUI	6509 (100.00%)	95.06	96.67 (96.67)	95.86 (95.86)
has_PE	umlsPF:::SNOMEDCT, wikiCategory, pairwiseLinkFeatures, sectLinkSectPath	26 (0.25%)	100	100 (0.25)	100 (0.5)
induces	umlsPF:::SNOMEDCT, umlsSemType, umlsCUI	194 (71.59%)	92.22	85.57 (61.25)	88.77 (73.61)
may_diagnose	umlsPF:::SNOMEDCT, umlsSemType, umlsCUI	132 (34.20%)	85.47	75.76 (25.91)	80.32 (39.76)
may_prevent	umlsPF:::SNOMEDCT, umlsCUI	598 (67.80%)	84.92	71.57 (48.53)	77.68 (61.76)
may_treat	umlsPF:::SNOMEDCT, umlsSemType, umlsCUI	10003 (76.11%)	88.6	93.97 (71.53)	91.2 (79.15)
NOREL	umlsPF:::MSH, umlsSemType, umlsCUI	12918 (26.62%)	97.78	95.7 (25.48)	96.73 (40.43)

Table 6.3: Performance of best feature sets per relation on test instances covered by a specific feature set. R and F_1 for the same feature set on the full data set are reported in parentheses (P remains the same under both conditions).

of sentences. We queried PubMed for abstracts and titles containing pairs of entities from our dataset using *NCBI Entrez Utilities Web Service*[8], and labeled sentences containing e_1 and e_2 with relation R. Overall we have extracted 122,466 sentences for the entity pairs from the *AUTONDF* dataset. These sentences were then used to train a system to predict relations between entities in the context of a sentence. We used features motivated by lexical features presented in [Mintz et al., 2009]. Specifically, we used word lemmas and part of speech tags of three words to the left and right of both entities, word lemmas between the entities and a binary feature denoting which entity comes first in the sentence. In addition, we also used the distance between both entities in terms of words.

In testing phase, to predict the relation between an entity pair, we used the majority prediction by this system on the set of all sentences extracted for that pair from PubMed. The baseline system is implemented using the multi-class linear kernel support vector machine (SVM)[Cristianini and Shawe-Taylor, 2000] classifier. More specifically, we used *libsvm* library[C. and L., 2001].

6.4.3 Entity-level semantics (ELS) systems

Each individual ELS system is a linear SVM classifier operating upon a vector of a subset of features described in Section 6.3. The only difference between different individual systems is the feature set employed. We created 49 individual systems, based on different feature type combinations. The combinations with very small coverage are not considered. A feature set coverage is considered too small if the corresponding covered subset of *AUTONDF* did not contain enough instances to carry out a reliable evaluation.

[8] `http://www.ncbi.nlm.nih.gov/entrez/query/static/esoap_help.html`

Results Performance is evaluated using 10-fold cross-validation on the *AUTONDF* dataset. Results over individual folds are averaged in order to obtain the results over the entire dataset. We report the performance of our systems in terms of precision (P), recall (R), and F_1 measure (F_1). Here we use standard formulas for P, R and F_1 in multi-class setting.

Table 6.2 reports the performance of the distant supervision baseline for each relation type. Here, precision, recall and F_1 are calculated over the instances for which the classifier is able to make a prediction (instances not covered by the classifier are skipped from evaluation). F_1 and recall over the full dataset are reported in parentheses, precision remains the same under both conditions. The column *Count (Coverage)* reports the number and percentage of entity pairs of a relation type for which the classifier is able to find sentences and create instances. First, we can see that the coverage of DS is rather poor. Due to this, the classifier is not able to learn reliable models in many cases (e.g. CI_MoA, and induces). There is only one relation, may_treat, for which the classifier finds adequate number of instances for training, resulting in a reasonable F_1. We also evaluated the baseline on the entire dataset (the table is not shown due to space limitations). The precision remains the same (as the number of instances retrieved does not change with the evaluation set), but the recall numbers drop drastically, resulting in very poor F_1s. In spite of using a rich resource such as Pubmed, we found that this classifier faces coverage issues because first, not all relations of interest are commonly expressed in sentences, and second, not every entity pair, from our large entity pair dataset, always co-occurs in sentences.

Table 6.3 reports the performance of our ELS classifiers. Again, performance is calculated for the covered instances. For space reasons, only the best performing classifier (based on F_1) is shown for each relation type. Note that Overall numbers are not shown in this table as these are differ-

ent classifiers. The second column (Best FS) reports the feature set of the best-performing classifier and the third column reports its coverage. First, we notice that the coverage of these classifiers are much higher than DS for most relation types (except for has_PE). Specifically, there is a 44 percentage point improvement on average. Second, the precision, recall and F_1s obtained by using entity-level semantics are substantially higher than that obtained by DS. Specifically, all F_1s are greater than 75%, and for six relations, the F_1 achieved is greater than 90%. On an average, this is a 39 percentage point improvement. This indicates that, for the detection of our medical relations, features using entity-level semantics is better than sentence-level features.

Observe that the best performing classifier is different for different relation types. For example, recognition of CI_MoA is most benefited by Wikipedia and entity-pair features, while may_treat is best benefited by category features from UMLS. Interestingly, observe that the very simple feature set (umlsSemType, umlsCUI) is the best performer for has_MoA. Additionally, by virtue of being available for all entities in our dataset, we also observed that this is the only feature set that has 100% coverage.

6.4.4 Ensemble of entity-level semantics classifiers

The previous subsection showed that systems using entity-level semantics have better performance than a system using sentence-level information. We also saw that systems with complex features may suffer from coverage issues while systems with simple features may not be discriminative enough. However, due to the difference in coverage and performance for different relation types, it is difficult to select one universally best system. Further, in many cases, a new instance to be classified has coverage in more than one ELS system, and it is difficult to decide which system's prediction is to be considered.

In order to get the best in terms of performance as well as coverage, we combine all 49 ELS systems in a *ensemble* system which takes outputs of individual systems for an instance as input, and predicts a single relation class label.

The ensemble classifier has a feature corresponding to each ELS classifier. Given an entity pair, the feature value for an ELS classifier feature will be its relation prediction for that entity pair (or "notCovered" if there in no coverage for that classifier). This classifier is also implemented using libsvm.

Results Table 6.4 reports the performance of the ensemble classifier on the entire *AUTONDF* data. Ensemble classifier for i-th test fold of cross validation was trained on the outputs obtained by the individual classifiers on $1, 2, i-1, i+1, 10$-th test folds. For comparison, we also report the performance of the best ELS system that has full coverage, STCUI. STCUI uses only the simple semantic features: Umls semantic type, and CUI. Here we see that in addition to full coverage, the ensemble is also able to achieve better performance than STCUI for all relation types. The improvement in F_1 is due to the improvement in both precision and recall. The improvements that are significant at $p < 0.01$ are shown in bold. Thus, by combining the individual ELS classifiers, it is possible to harness different types of entity-level semantics to achieve good coverage as well as performance for relation mining.

6.5 Discussion

UMLS semantic type has been frequently used as one of the most useful semantic features[Abacha and Zweigenbaum, 2011b]. However, we found that, these features can be too coarse to be discriminative for our task.

Relation	Ensemble			STCUI		
	P	R	F1	P	R	F1
CI_ChemClass	**93.47**	92.54	**93**	91.56	93.62	92.58
CI_MoA	**93.25**	95.6	**94.41**	88.56	94.97	91.65
CI_with	**93.67**	**94.49**	**94.08**	92.45	93.58	93.01
has_Ingredient	**78.11**	**63.25**	**69.9**	73.11	53.87	62.03
has_MoA	**95.26**	**97.05**	**96.15**	95.06	96.67	95.86
has_PE	95.82	96.5	96.16	95.76	96.53	96.14
induces	91.67	81.18	**86.11**	88.26	74.91	81.04
may_diagnose	89.91	**76.17**	**82.47**	87.77	72.54	79.43
may_prevent	**81.45**	**68.71**	**74.54**	77.13	54.31	63.74
may_treat	**90.57**	92.48	**91.51**	87.51	92.2	89.79
NOREL	**96.49**	**96.38**	**96.43**	96.33	95.7	96.01
Overall	**90.88**	**86.76**	**88.61**	88.50	83.54	85.57

Table 6.4: Performance of ensemble and STCUI baseline systems. Overall is obtained by macro-averaging over results for individual relations.

For instance, consider the entity pair *Secretin* (e_1)- *Liver Diseases* (e_2). UMLS semantic types of an entity e_1 was found to be *Hormone*, *Pharmacologic_Substance*, *Amino_Acid,_Peptide,_or_Protein*, while e_2 has semantic type *Disease_or_Syndrome*. On the other hand SNOMED CT contains information that e_1 is *Gastrointestinal hormone* and *Peptide hormones and their metabolites and precursors*, while e_2 is a *Liver finding*, *Disorder of abdomen* and a *Disorder of digestive organ* . Intuitively such fine-grained information is more discriminative. Table 6.3 corroborates this intuition – most of the top classifiers that use entity category information infact make use of SNOMED CT features.

The semantic features we employ vary from simple identity-based features such as umlsCUI, to complex pair-based features such as pairwiseLinkFeatures. Entity pair features are complex, and relatively sparse, which makes learning them reliably a challenge. However, the information they

capture can lead to creating more precise predictions. On inspecting instances that were incorrectly classified by classifiers using only simple category features such as umlsCUI, but correctly classified using pairwiseLinkFeatures features, we found that the classifier with only identity-based features predicted the most common relation that the given entities were involved in, while the classifier incorporating pairwiseLinkFeatures overcame this pitfall.

Finally, we experimented with extending the sentence-level baseline classifier, DS, with the simple semantic features. Here we augmented the existing feature vectors constructed using linguistic features with semantic information such as umlsSemType and umlsCUI. This approach does not change the coverage of DS, but allows us to inspect the impact on precision due to entity-level semantics. Table 6.5 reports the results, similar to Table 6.2. Here we see that, with the addition of even simple entity-level semantics, not only has the precision for most relations improved, but the recall of the relation types are improved as well, resulting in much higher F_1. Addition of more complex entity-level semantics and combining the sentence-level system with semantics-based system are directions for our future explorations.

6.6 Related work

Biomedical relation extraction Approaches to relation extraction in the biomedical domain include pattern based approaches [Abacha and Zweigenbaum, 2011a, Sahay et al., 2008, Ramakrishnan et al., 2006], machine learning approaches [Rosario and Hearst, 2004, Frunza and Inkpen, 2010, Shawe-Taylor and Cristianini, 2004, Giuliano et al., 2006, Li et al., 2008] or a combination of the two [Abacha and Zweigenbaum, 2011b]. For example, [Abacha and Zweigenbaum, 2011a] use a set of relation-specific

Table 6.5: Distance Supervision - Using STCUI

Relation	DS + STCUI		
	P	R	F_1
CI_ChemClass	72.96	84.06	78.11
CI_MoA	0.00	0.00	0.00
CI_with	89.77	60.11	72.01
has_Ingredient	82.76	37.50	51.61
has_MoA	78.18	89.58	83.50
has_PE	96.23	87.18	91.48
induces	87.10	45.00	59.34
may_diagnose	100.00	8.33	15.38
may_prevent	84.51	32.79	47.24
may_treat	79.51	97.67	87.66
NOREL	83.58	70.68	76.59
Overall	77.69	55.72	60.27

patterns, Sahay et al. [2008] use a set of syntactic patterns, and Ramakrishnan et al. [2006] extract relations matching a set of manually designed rules using an enriched syntactic parse tree representation of sentences. Our focus in this work is on supervised methods.

Supervised statistical machine learning (ML) approaches automatically learn patterns in the labeled data. Rosario and Hearst [2004] recognize *disease*, *treatment* semantic role and seven semantic relations, and extract 7 binary and unary relations between them: *cure, only DIS, only TREAT, Prevent, Vague, Side Effect, NO Cure* using discriminative models. Frunza and Inkpen [2010] distinguish between three relation classes, *cure*, *prevent* and *side-effect*, experimenting with various feature representations. The best results were achieved using rich feature sets (bag of words, noun phrases, verb phrases, UMLS semantic types). The authors mention that better results are achieved when ontological knowledge is employed. We too use a supervised setting, and some of our semantic features overlap with

theirs. However, our work focuses on exploring an assortment of semantic features alone, as sentences and consequently sentence-based features have low coverage for our task. Our results corroborate that semantic features are important for relation extraction in this domain. However, we focus on a different set of biomedical relations from the above. Additionally, we show that using classifier ensembles can overcome the difficulties due to lack of coverage.

Relation extraction using semantic knowledge. In the biomedical domain semantic knowledge is exploited previously by Rosario and Hearst [2004] who used MeSH IDs of the words occurring in a sentence being classified as features. UMLS features have been added to sentence-level features in relation mining with promising results in Frunza and Inkpen [2010] and Abacha and Zweigenbaum [2011b]. We found that sentence-based systems have poor coverage for our task, which we remedy using a variety of semantic information and then fusing them.

Distant supervision for relation mining. Distant supervision (DS) approaches for relation mining have used Freebase[Mintz et al., 2009] and YAGO[Nguyen and Moschitti, 2011] to extract labeled sentences from Wikipedia. Yao et al. [2010] use an undirected graphical model for both relation and entity type prediction and use Freebase as a source of seeds, and Wikipedia and New York Times corpus as source of sentences. The problems of DS approaches are the noise in the data and absence of knowledge about negative instances and their distribution. Moreover, in our task, the sentence retrieval lacks coverage.

Ensemble Learning. Ensemble learning methods have been applied to a variety of natural language processing applications such as those for text

categorization [Sebastiani, 2002], parsing [Collins and Koo, 2005], word-sense disambiguation [Pedersen, 2000, Escudero et al., 2000]. In relation mining, they have been used for ontology learning within a system called OntoLancs [Gacitua and Sawyer, 2008]. [Van Landeghem et al., 2010] use ensemble feature selection for biomolecular text mining. They show that their feature selector is able to discard a large fraction of machine-generated features, improving classification performance of state-of-the-art text mining algorithms. While we use an ensemble approach, the main focus of our work is on exploration of a variety of entity-level semantics for detecting different clinical relations.

6.7 Conclusion

This work explored use of rich knowledge about biomedical entities obtained from various sources for relation mining. Our entity-level semantics includes taxonomic information about individual entities as well as linkage information between entity pairs. We built individual classifiers that harness entity semantics as well as a meta classifier to achieve advantages of performance and coverage. Our approach was tested on a large dataset obtained from a standard human-curated ontology.

Our experiments reveal that the distant supervision approach that uses sentence-level information does not perform well for our domain and relation types – it has issues with both coverage and performance. We discovered that different types of semantics are useful for different relation types, and that performance and coverage vary based on the scope and depth of the knowledge sources used. Our ensemble approach proved successful in solving the problem of coverage, while achieving good overall performance.

Chapter 7

Improving linking to Wikipedia

In this chapter of we propose a methodology for improving the accuracy of linking terms in a plain text to Wikipedia pages. The approach is based on applying the one sense per discourse hypothesis to Wikipedia pages and categories in order to automatically expand Wikipedia annotations. Experiments show that the hypothesis is generally correct within Wikipedia allowing us to improve disambiguation accuracy on a benchmark data set.

7.1 Introduction

Wikipedia has been successfully used to extract training data for supervised word sense disambiguation (WSD) systems [Csomai and Mihalcea, 2008]. The huge and continuously growing amount of training data that the free online encyclopedia makes available has allowed supervised approaches to regain popularity. This is because, in spite of the F_1 of the best supervised systems is around 73% (Senseval-3, Task 3, [Mihalcea and Edmonds, 2004]) and 82.5 - 88.7% (SemEval-2007, Task 07 [Navigli et al., 2007], Task 17 [Pradhan et al., 2007]), respectively, in fine- and coarse-grained evaluations, they were not applicable in practical applications for the high cost to create and maintain the training data. For this reason, in the last years

unsupervised techniques were preferred to supervised. Until recently their performance had been unsatisfactory, typically few points above a baseline that selects the most frequent word sense by default. Recently, unsupervised knowledge-based WSD has benefited from merging Wikipedia and WordNet into a large-scale semantic network, BabelNet [Navigli and Ponzetto, 2012]. Knowledge-based unsupervised WSD methods exploiting BabelNet achieve performance comparable to that of the supervised systems [Ponzetto and Navigli, 2010]. We believe that Wikipedia and other collaborative resources could similarly be beneficial for the performance of the supervised techniques by helping to overcome the labeled data bottleneck.

Wikipedia supplies senses for a large number of words and, for each sense, frequently, labeled examples to train a word expert classifier [Mihalcea, 2007]. Specifically, word senses are represented by Wikipedia articles and their labeled examples are obtained from the articles in which the word occurs as an anchor text of an internal hyperlink. For example, the word *rally* has two frequent senses, *Demonstration_(people)* and *Rallying*, for which we can collect 74 and 725 training examples, respectively. Despite this technique mainly applies to nouns, it provides the largest training set available for WSD.

However, the problem cannot be considered solved yet as the distribution of the Wikipedia annotation is highly skewed and consequently many word expert classifiers show significantly lower performance when tested outside Wikipedia. In particular, few words have a large number of examples, while the majority have a small number. Frequently, rare senses have a lot of examples, e.g., *heel* has 74 training examples as "the body part" and 733 as "a contemptible character of professional wrestling". Furthermore, the Wikipedia contributors are recommended not to link common

words (e.g., state), in order to avoid over-linking.[1] Therefore, Wikipedia pages about non-domain specific concepts are infrequently linked to. Additionally, most frequent senses are sometimes missing, e.g, *head* has no examples as "person in charge of something," while define specific concepts, e.g., *head of state* or *head of department* might be present.

In this chapter we aim to increase the amount of labeled data obtained from Wikipedia internal links thus reducing skewness of sense annotations towards domain-specific senses. For this purpose we investigate the use of the one sense per discourse hypothesis [Gale et al., 1992, Yarowsky, 1995] applied to Wikipedia data. Specifically, we apply the hypothesis to Wikipedia articles and categories, and evaluate the amount of fresh training data we can derive and its impact on the performance of a supervised WSD system. The results show that the hypothesis is correct for articles, but precautions have to be taken to ensure that this assumption does not lead to erroneous discoveries when categories are considered.

7.2 Adapting the One Sense Per Discourse hypothesis to Wikipedia

The *one sense per discourse* hypothesis states that all occurrences of a word within the same discourse tend to share the same sense [Gale et al., 1992, Yarowsky, 1995]. If the hypothesis holds, it is extremely likely that all occurrences of a polysemous word within the same article will share the same sense. [Gale et al., 1992] found in their experiments that the tendency to share sense in the same discourse is around 98%. For example, all the 100 occurrences of the word *virus* in the pages categorized as `Computer_security_exploits` share the same sense, but only 4 occurrences are linked to the article `Computer_virus` and, consequently, ex-

[1] `http://en.wikipedia.org/wiki/Wikipedia:Tutorial/Wikipedia_links`

ploited as training data. In the extreme case, in which a sense has no examples at all, a small number of annotated words could be provided to bootstrap the process, e.g., by exploiting crowdsourcing.

It has been shown that the validity of the hypothesis heavily depends on the granularity of the sense inventory. Experimenting on SemCor,[2] Krovetz [1998] found significantly more occurrences of multiple-senses per discourse than reported by Gale et al. [1992], e.g., more than 40% for nouns. The significant difference between the two outcomes is determined by the more fine-grained sense distinctions considered by Krovetz [1998]. Therefore, the fine-grained sense distinctions present in Wikipedia [Mihalcea, 2007, Wolf et al., 2010], even though in part due to presence of named entities, hypernyms, and hyponyms among the word senses, seem to suggest that we cannot indiscriminately adopt the one sense per discourse hypothesis. These considerations motivate our investigation.

In the context of Wikipedia we convert the one sense per discourse hypothesis to *one sense per article* and *one sense per category* hypotheses.

One sense per article (OSA). The *one sense per article* hypothesis states that if a word in a Wikipedia article has been annotated with a link to another Wikipedia article, i.e. labeled, then the label can be propagated to all the unlabeled occurrences of the word within the article. The only exception to this hypothesis is when a word is labeled with two or more different labels within the same article. In this case the hypothesis does not hold and the propagation cannot be performed.

Following the hypothesis we wrote a *in page* label propagation procedure that propagates the word sense to multiple occurrences of the word in the page. If case when a word has multiple senses (is linked to different pages) in the page, we did not propagate.

[2] `http://www.cse.unt.edu/~rada/downloads.html#semcor`

One sense per category (OSC). The *one sense per category* hypothesis makes similar to OSA assumptions about labeled words and Wikipedia categories. Wikipedia categories indicate the topic or subject of a page, and the Wikipedia guidelines suggest to assign the most specific categories to the pages.[3]

Label propagation within Wikipedia categories is a more problematic issue as compared to OSA and it needs particular attention. Not all the categories are guaranteed to be strongly topically coherent and therefore indicative of a word sense. Intuitively, the larger the category is, the less likely is that it is strongly topically coherent. For example, `Living_people`[4] lists 597,678 pages, while `1847_births` lists 1,253 pages. In case of such categories, even if an ambiguous noun is not annotated with contradicting labels within a category, we are not guaranteed that OSC holds, and consequently risk to obtain a large amount of wrongly labeled examples as a result of propagation. This motivates us to assume that the hypothesis does not hold for extremely populated categories, while it is more realistic that it holds for the less populated ones. Recall that our goal consists in maximizing the number of new labeled examples and minimizing the noise, thus we need to find a trade-off between the two dimensions. We devised an *in category* label propagation procedure, which is requires all the labeled occurrences of a word within a Wikipedia category should share the same label. Additionally, because of considerations listed above, we impose a restriction on the number of pages that populate a specific category.

The label propagation procedure based on these considerations is outlined as follows:

For each page p_0 in which a noun n occurs with label l, we retrieve

[3] `http://en.wikipedia.org/wiki/Wikipedia:Categorization`

[4] Original category name may be induced by adding prefix `http://en.wikipedia.org/wiki/Category:`

all pages p_i $(i > 0)$ that are contained in the category C_j $(0 < j < M)$ such that $p_0 \in C_j$ and $|C_j| < v$. Then, if n has no multiple-senses in $C_j = \{p_0, p_1, \ldots\}$, we propagate the label l to all unlabeled occurrences of n in C_j.

Even though this latter assumption is not accurate, a label propagation procedure based on it allows us to limit the search space, otherwise for each example we would obtain too many pages.

7.3 Experiments

We conducted experiments to determine (i) how often nouns have more than one meaning per discourse within Wikipedia, (ii) the impact of the training data obtained by using the different Wikipedia adaptations of the one sense per discourse hypothesis on the performance of a supervised word sense disambiguation system. Additionally, for each hypothesis we show how many new examples we are able to collect for increasing samples of existing labeled examples. We estimate the level of noise in the training data obtained by using the OSA or OSC procedures. We call an article or a category "noisy" if it contains two or more inter-article links with the same anchor text but with different targets.

However, these methods could underestimate the correct number of occurrences of multiple-senses per discourse, and the noise introduced could be higher than the one estimated as typically few words are linked. For this reason, we also perform an indirect evaluation by comparing the performance of *The Wiki Machine* (TWM) (see Chapter 3) on Wikipedia before and after the label propagation.

7.3.1 Test set

In order to evaluate our methodology we need a dataset (i) annotated with links to Wikipedia; (ii) with sense distribution not skewed towards domain-specific senses. We cannot run experiments on native Wikipedia data as they would suffer from skewness of sense distribution towards named entities and domain-specific senses. Moreover, none of the publicly available word sense disambiguation datasets annotated with links to Wikipedia fully fits our requirements. First, in some datasets the prevailing amount of annotations are those of named entity mentions or domain-specific words [Milne and Witten, 2008]. In the other datasets the annotations were not bound to specific occurrences in a context, but rather to an entire document [Mendes et al., 2011, Cucerzan, 2007], while we would like to take into account each occurrence of an ambiguous word in evaluation. Conventional WSD evaluation datasets, such as SemEval and SensEval, do not suffer from such problems, but there exists no manually elaborated mapping between Wikipedia and WordNet.

After considering the possible options we decided to produce the dataset manually. In order to avoid substantial effort on annotation, we take a conventional dataset annotated with WordNet senses, and manually map the WordNet synsets to Wikipedia pages, similarly to [Mihalcea, 2007].

The experiments have been conducted on 57 polysemous nouns,[5] that is a randomly selected subset of nouns employed in the Senseval/Semeval evaluations. The selected nouns have average polysemy and average number of labeled examples of 22 and 1,911 respectively in the April 2010 Wikipedia version. Their average polysemy is 6 in WordNet. We extracted

[5]Antenna, arm, atmosphere, audience, bank, bass, bow, campaign, cancer, cone, crane, degree, deposit, difference, difficulty, disc, drill, drug, drum, duty, galley, hull, image, interest, interior, issue, jaguar, judgment, knife, land, landscape, language, leopard, line, marine, mole, organ, paper, park, party, performance, position, rally, scale, sentence, shelter, slug, sort, source, star, table, taste, tiger, tree, trunk, and virus.

the test examples from the English SemCor corpus. Originally they are annotated with WordNet senses, while in our study we exploit the Wikipedia articles as a sense repository. One annotator mapped the WordNet senses of the test nouns to corresponding Wikipedia pages. We did not use any of the available automatic Wikipedia-WordNet alignments, as we do not want the possible noise in these alignments to influence the experiment results. The automatic alignment provided by [Navigli and Ponzetto, 2010] has been used for comparison purposes only.

Aligning WordNet synsets with Wikipedia pages which correspond to their hyponyms was not allowed. Without this restriction the link-based training set for a high-level generic sense of a noun might be biased towards the domain of a specific hyponym(s). For instance, WordNet sense `shelter#1`, *"a structure that provides privacy and protection from danger"*, was not mapped to the Wikipedia pages `Animal_shelter` and `Women's_shelter` as they are hyponyms. Overall, the test nouns correspond to 342 WordNet synsets. We found a corresponding Wikipedia page for 174 (around 50%) of them.

The test set contains only the examples with the sense labels for which we found a corresponding Wikipedia sense. In case if multiple sense labels were mapped to the same Wikipedia pages, we did not include examples annotated with these sense labels into the test set, as they might result in more coarse-grained sense annotation than the one originally present in SemCor. The final test set consists of 878 examples for 41 nouns of interest[6] labeled with 68 senses out of those 174 senses for which WordNet-Wikipedia mapping exists. The actual number of senses in the final test corpus is smaller than the one in the WordNet-Wikipedia mapping, because not all of the senses for which the mapping was found have examples

[6]This number is smaller than the original number of nouns selected, due to the fact that for certain nouns none of their senses were mapped to Wikipedia

in SemCor. For example, we mapped `bass#8` WordNet sense, "nontechnical name for any of numerous edible marine and freshwater spine-finned fishes," to the Wikipedia page `Bass_(fish)`, however, SemCor does not contain any examples for this sense of "bass."

Note that, even though, after removing the non-mapped senses the reported ambiguity in the test set has reduced, the average ambiguity in the training set extracted from Wikipedia still remains 22 as reported before. Moreover, the most frequent senses in the training and test sets do not match. For example, the most frequent sense of "arm" in the test set is `Arm`[7] while in the training set it is `Coat_of_arms`.

7.3.2 One sense per article procedure evaluation

We identified all articles in which a noun occurs linked to more than one target page (sense). The 57 nouns occur 279,151 times in 78,469 pages, among these 83,515 occurrences are labeled. The one sense per article hypothesis is violated, i.e. a noun is annotated with two different labels within the same page, in only 0.76% of the pages. Figure 7.1 shows how many new examples we are able to collect for increasing samples of existing labeled examples; the average ratio between acquired and existing examples is ~ 2.5.

The indirect evaluation of the amount of noise in the collected data was performed in the following way. We created 3 disambiguation models trained on the existing labeled examples (L), the examples extracted by the label propagation procedure (P), and a combination of the first two (LP). To conclude that the hypothesis holds, P must obtain results comparable with L. Figure 7.2.a compares the performance of the 3 models for different amount of labeled training data. The accuracy of the most frequent sense

[7]The most frequent sense of "arm" in WordNet and Semcor is that of a "human limb". It corresponds to the Wikipedia page, `http://en.wikipedia.org/wiki/Arm`

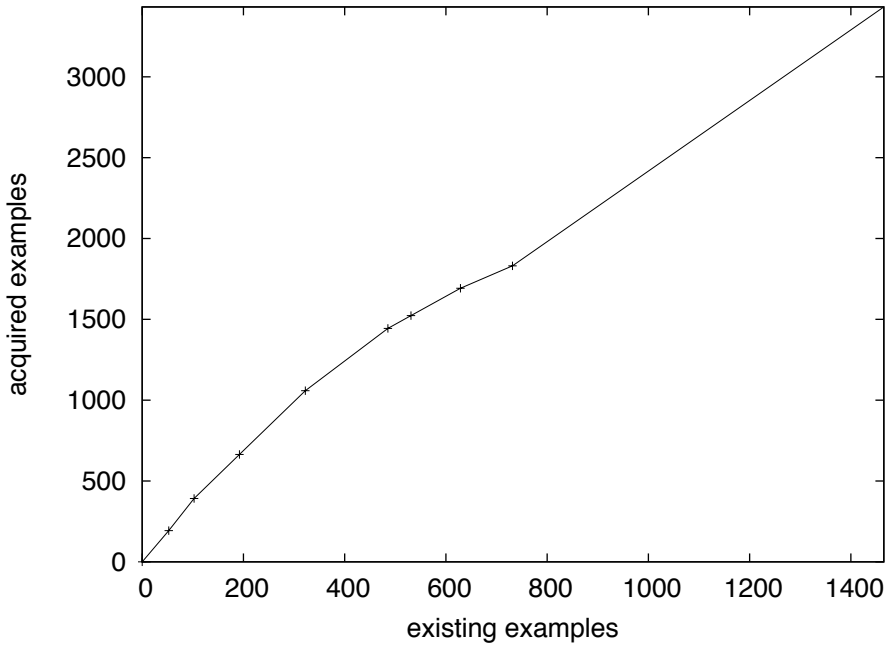

Figure 7.1: The ratio between the number of acquired and existing labeled examples.

baseline is 47%, and as the figure shows it is outperformed by all the models.

7.3.3 One sense per category procedure evaluation

By arbitrarily setting $v = 20$, we collected 127,117 new examples from 15,898 categories. The hypothesis is violated, i.e. a noun is annotated with contradicting links within the same category, in 2.1% of the categories. Smaller v would result in fewer propagations, while too large v would increase the probability of the hypothesis violation and result in a large amount of noisy examples.

We have used the examples obtained by the label propagation procedure applied to categories to train 2 additional disambiguation models, one from acquired examples only (C) and the other from the combination with label examples (LC). Figure 7.2.b compares the performance of the models for

different amount of labeled training data.

7.4 Discussion

As expected, label propagation introduces a certain amount of noise as the one sense per discourse hypothesis is not completely correct. However, all experiments we performed confirm its general validity within Wikipedia, even though the results presented by [Krovetz, 1998] could suggest the contrary since the sense inventory derived from the online encyclopedia provides fine-grained sense distinctions [Csomai and Mihalcea, 2008, Wolf et al., 2010]. The small difference in performance between the disambiguation models L and P further confirms that the real error rate is approximately the one estimated using the hyperlinks only ($\sim 1 - 2\%$).

On the other hand, the acquired training data provide additional information, allowing the combined models LP and LC to outperform the basic models L, P and C. As expected, the improvement is more significant for small amount of training data. The model C shows significant variation in performance for different amount of training examples but, interestingly, it also shows the highest accuracy with just $\sim 50\%$ of the training examples. This is probably due to our in category label propagation procedure that strongly depends from the seed examples used. This suggests that a appropriate technique for selecting the categories where to propagate the labels could improve the performance, as sampling examples from different pages maximizes the diversity between training examples.

Performing the mapping between WordNet and Wikipedia, we discovered that we could not map 53% WordNet synsets corresponding to our nouns of interest to Wikipedia pages, due to the absence of the latter. In most of the cases mappings are missing for the common non-domain-specific senses. Consequently, we were not able to collect training data for

them. Such generic senses are very frequent in text and this result must be taken into account when building a disambiguation system based on Wikipedia. On the other hand, for most specific senses Wikipedia is certainly richer than WordNet. This confirms the hypothesis that Wikipedia and WordNet have complementary sense repositories [Wolf et al., 2010, Navigli and Ponzetto, 2010]. The lack of generic concepts could be partially due to method we use to create the sense inventory, in which all possible senses of a word are determined by the pages it links to.

7.5 Conclusion

This chapter describes an adaptation of the one sense per discourse hypothesis to the Wikipedia structure, giving a positive answer to the question "Does the One Sense per Discourse Hypothesis hold within Wikipedia?." It explores the validity of the hypothesis within Wikipedia articles and categories. The results obtained show that this direction is promising but a more stable propagation procedure must be found. Finally, we have shown that label propagation based on all the adaptations of this hypothesis allows improving the accuracy of WSD based on Wikipedia.

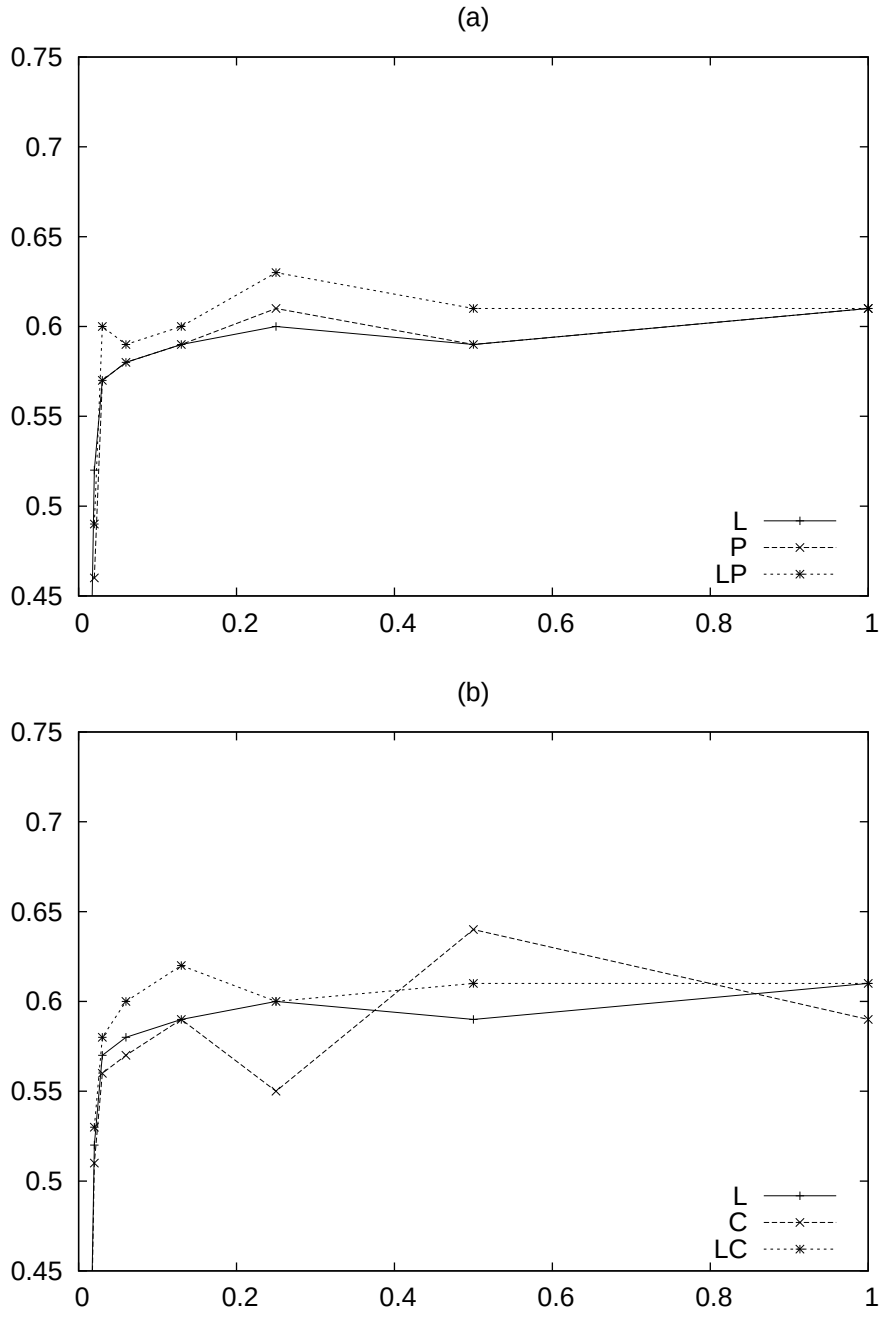

Figure 7.2: The performance comparison between the different disambiguation models: (a) in page label propagation procedure, (b) in-category label propagation procedure. Ace Y corresponds to the accuracy; ace X corresponds to the fraction of labeled training data used for the propagation propagation procedure.

Chapter 8

Conclusions

In this thesis we have proposed a generic framework for using background knowledge from Linked Open Data (LOD) in Natural Language Processing (NLP) tasks. The framework consists in (i) mapping terms in source text to Wikipedia pages, (ii) using Wikipedia page names of these pages as a mediator to obtain knowledge from LOD resources, (iii) converting this knowledge into the features and injecting them into machine learning algorithms.

We have provided recommendations for the practical implementation of the constituents of the framework, including linking to Wikipedia and organizing LOD knowledge extraction. We have developed and described *j-lod-feature*, a tool for extracting relevant portions of LOD knowledge given a Wikipedia page name, and converting it to a predefined, but extensible set of features.

We have proposed a methodology for improving the performance of TWM. Since TWM is a supervised system that exploits the internal Wikipedia links to create training data, we aim to increase the amount of such links. We create new links using existing links and adapting the *one sense per discourse* hypothesis to Wikipedia pages and categories. Our experiments indicate the overall validity of the hypothesis. The future work in this di-

rection includes further refining of the hypothesis adaptation for categories, so that it would result in smaller amount of noise.

We have evaluated the applicability and the performance of the framework on examples of three case-studies: coreference resolution, semantic relation extraction and relation mining in the biomedical domain.

Coreference resolution. In coreference resolution case-study we extracted background knowledge about entity mentions from DBpedia, Freebase and YAGO, converted it into features and applied a feature selection method for selecting a task-relevant subset. We have injected the features into a knowledge-lean machine learning system, implemented as a Markov Logic Network. We have observed that LOD-based semantic features results in increase of recall and F_1-measure.

Note that this research direction is evolving, and after our publication [Bryl et al., 2010] there were other works further investigating related ideas [Rahman and Ng, 2011, Ratinov and Roth, 2012].

Semantic relation extraction. We have applied the framework to the task of semantic relation extraction between pairs of nominals. We experimented with WordNet, OpenCyc, YAGO and their combinations, as sources of background knowledge. We compared word sense disambiguation through Wikipedia mediation to the baseline approaches, such as the *most frequent sense* approach and *all senses* approach.

We have discovered that, first, usage of Wikipedia as a semantic mediator is problematic at the current stage of development of LOD and TWM, and is outperformed by the baseline disambiguation strategies. We have presented the detailed error analysis of the reasons. Second, we have shown that even with the disambiguation step omitted the combination of WordNet and OpenCyc knowledge with shallow syntactic features results in the state-of-the-art performance comparable

to that of the top-performing SemEval-2010 system.

Biomedical relationship mining. We have investigated the task of relationship mining between pairs of biomedical entities, such as drugs, diseases, methods of actions, physiological effects and chemical classes. We have observed that substantial amount of relevant information was not present in Linked Open Data or present only partially at the moment of conducting the investigation. Therefore, in order to achieve maximal performance possible, we have exploited background knowledge that is also unavailable in LOD. More specifically, we have used UMLS Semantic Network, MEDCIN, MeSH and SNOMED CT, Wikipedia as sources of background knowledge. Here MEDCIN and SNOMED CT are proprietary resources not available on LOD.

We have built a set of individual classifiers exploiting different combinations of semantic features extracted from the above-mentioned knowledge sources. We have shown that different kinds of semantic features, incorporated in different classifiers are relevant for different relation types. Note that these classifiers all have different coverage, depending on the feature sets that they employ. Finally, we have demonstrated that an ensemble approach, that combines the predictions of individual classifiers, helps to improve the overall performance of the relation mining system and to increase its coverage.

Summarizing the insights from our case-studies we can state that:

1. Framework performs well in case when semantic features are extracted for named entity mentions, due to the fact that they are well represented in Wikipedia which we use as a semantic mediator.

2. Framework may encounter problems due to the

- absence of Wikipedia pages describing very common concepts. Partially, this could be remedied over time when corresponding pages appear on Wikipedia. On the other hand, Wikipedia has encyclopedic nature, and some very high-level common domain concepts, e.g. "configuration", are not likely to appear there as distinct articles. Therefore, it might be reasonable to look for an additional mediator.
- Missing and noisy links between the distinct LOD resources. They result in missing and noisy data. This can be remedied over time as LOD develop. In 2010, a new four-year LOD2[1] project was launched within Seventh Framework Programme aiming to develop new LOD2 technologies, including the technologies for high-quality interlinking.

The aforementioned problems are especially relevant for the cases when the terms of interest are common nominals.

3. While LOD have high coverage in the general-purpose domain, some important resources from the specialized domains, such as biomedical domain, are not yet available there. This can change over time when new datasets are added to LOD. So far the number of datasets in LOD has been rapidly growing, their number has evolved from 12 datasets in 2007 to 95 in 2009 and 295 in 2011.[2]

4. We have shown that SW tools, LOD architecture and RDF data representation format allow us to reduce the technical effort when experimenting with semantic data from different sources. For instance, without SW, LOD, and RDF in semantic relation extraction case-study we would have to study and employ completely different APIs

[1] `http://lod2.eu`

[2] According to `http://richard.cyganiak.de/2007/10/lod/`

when extracting knowledge from WordNet and OpenCyc. SW enables us to build a tool, that would extract semantic features from sources made available by different owners in a uniform manner.

5. Confirming observations by Mihalcea [2007] we can state that even though Wikipedia provides a very large amount of labeled data for word sense disambiguation, sense distribution in these data is skewed towards domain-specific and named entity senses, while more general senses are underrepresented. As we show in Chapter 7, this can be partially overcome by automatically propagating links within categories and pages in which different occurrences of an ambiguous term tend to exhibit the same sense.

Future work directions include (1) further improving the methodology of mapping plain text to Wikipedia, as this is a bottleneck for the framework performance; (2) looking for alternative ways of mapping to LOD sources in cases when Wikipedia page for a given concept is not available; (3) testing the framework on other tasks; (4) defining new LOD-based features.

Bibliography

Semantic web vocabularies reference, 2012. URL `http://www.w3.org/standards/semanticweb/ontology`.

A.B. Abacha and P. Zweigenbaum. Automatic extraction of semantic relations between medical entities: a rule based approach. *Journal of Biomedical Semantics*, 2(Suppl 5):S4, 2011a.

A.B. Abacha and P. Zweigenbaum. A hybrid approach for the extraction of semantic relations from medline abstracts. In *In Proceedings of CICLing-2011*, pages 139–150. Springer-Verlag, 2011b.

ACE. Automatic Content Extraction. *http://www.ldc.upenn.edu/Projects/ACE/*, 2000-2005.

E. Bengtson and D. Roth. Understanding the value of features for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Association for Computational Linguistics, 2008.

L. Bentivogli, P. Forner, C. Giuliano, A. Marchetti, E. Pianta, and K. Tymoshenko. Extending English ACE 2005 Corpus Annotation with Ground-truth Links to Wikipedia. In *23rd International Conference on Computational Linguistics*, pages 19–26, 2010.

T. Berners-Lee. Linked data, 2006. URL `http://www.w3.org/DesignIssues/LinkedData.html`.

T. Berners-Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

C Bizer, J. Lehmann, G. Kobilarov, S. Auer, R. Becker, C.and Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, September 2009. ISSN 15708268.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

V. Bryl, C. Giuliano, L. Serafini, and K. Tymoshenko. Using background knowledge to support coreference resolution. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, August 2010.

R. Bunescu and R. Mooney. Subsequence kernels for relation extraction. *Advances in neural information processing systems*, 18:171, 2006.

R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*, volume 6, pages 9–16, 2006.

R.C. Bunescu and R.J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731. Association for Computational Linguistics, 2005.

Chih-Chung C. and Chih-Jen L. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

D. Carter. *Interpreting anaphors in natural language texts*. Halsted Press, 1987.

Y.S. Chan and D. Roth. Exploiting background knowledge for relation extraction. In *Proceedings of COLING-2010*, pages 152–160. ACL, 2010.

Y. Chen, M. Lan, J. Su, Z.M. Zhou, and Y. Xu. Ecnu: Effective semantic relations classification without complicated features or multiple external corpora. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 226–229. Association for Computational Linguistics, 2010.

N. Chinchor and L. Hirschmann. Muc-7 coreference task definition, version 3.0. In *Proceedings of MUC*, volume 7, 1997.

M. Collins and T. Koo. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–69, 2005.

N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines*. Cambridge University Press, March 2000.

A. Csomai and R. Mihalcea. Linking documents to encyclopedic knowledge. *IEEE Intelligent Systems*, 23(5):34–41, 2008. ISSN 1541-1672. doi: 10.1109/MIS.2008.86.

S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

A. Culotta and J. Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association*

for Computational Linguistics, pages 423–431. Association for Computational Linguistics, 2004.

A. Culotta, M. L. Wick, and A. McCallum. First-order probabilistic models for coreference resolution. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 81–88, 2007.

P. Denis and J. Baldridge. Joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 236–243, 2007.

P. Domingos, S. Kok, D. Lowd, H. Poon, M.Richardson, and P. Singla. Markov logic. In *Probabilistic Inductive Logic Programming*, volume 4911 of *Lecture Notes in Computer Science*, pages 92–117. Springer, 2008.

G. Escudero, L. Màrquez, and G. Rigau. Boosting applied to word sense disambiguation. *Proceedings of ECML-00*, pages 129–141, 2000.

C. Fellbaum et al. *WordNet: An electronic lexical database*. MIT press Cambridge, MA, 1998.

P. Ferragina and U. Scaiella. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628. ACM, 2010.

C.J. Fillmore, C.R. Johnson, and M. R. L. Petruck. Background to FrameNet. *International Journal of Lexicography*, 16:235–250, September 2003.

J.R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Pro-*

ceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 363–370. Association for Computational Linguistics, 2005.

O. Frunza and D. Inkpen. Extraction of disease-treatment semantic relations from biomedical sentences. In *Proceedings of BioNLP-2010*, pages 91–98. ACL, 2010.

E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1606–1611, 2007.

R. Gacitua and P. Sawyer. Ensemble methods for ontology learning - an empirical experiment to evaluate combinations of concept acquisition techniques. In *Proceedings of ICIS-2008*, pages 328 –333, 2008.

W. A. Gale, K. W. Church, and D. Yarowsky. One Sense Per Discourse. In *Proceedings of the 4th DARPA Speech and Natural Language workshop*, 1992.

R. Girju, P. Nakov, V. Nastase, S. Szpakowicz, P. Turney, D. Yuret, et al. Semeval-2007 task 04: Classification of semantic relations between nominals. *Urbana*, 51:61801, 2007.

C. Giuliano, A. Lavelli, and L. Romano. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proceedings of EACL-2006*, pages 401–408, 2006.

C. Giuliano, A. Lavelli, D. Pighin, and L. Romano. Fbk-irst: Kernel methods for semantic relation extraction. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 141–144. Association for Computational Linguistics, 2007a.

C. Giuliano, A. Lavelli, and L. Romano. Relation extraction and the influence of automatic named-entity recognition. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(1):2, 2007b.

C. Giuliano, A. M. Gliozzo, and C. Strapparava. Kernel methods for minimally supervised wsd. *Computational Linguistics*, 35(4):513–528, 2009.

R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. In *Proceedings of COLING*, volume 96, pages 466–471, 1996.

B.J. Grosz, S. Weinstein, and A.K. Joshi. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21 (2):203–225, 1995.

A. Haghighi and D. Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1152–1161, 2009.

A. Haghighi and D. Klein. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 385–393. Association for Computational Linguistics, 2010.

X. Han and L. Sun. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 945–954. Association for Computational Linguistics, 2011.

O. Hartig, C. Bizer, and J.C. Freytag. Executing sparql queries over the web of linked data. *The Semantic Web-ISWC 2009*, pages 293–309, 2009.

T. Heath and C. Bizer. Linked data: Evolving the web into a global data space. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 1(1):1–136, 2011.

I. Hendrickx, R. Morante, C. Sporleder, and A. van den Bosch. Ilk: Machine learning of semantic relations with shallow features and almost no data. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 187–190, 2007.

I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th SIGLEX Workshop on Semantic Evaluation*, pages 94–99, Uppsala, Sweden, 2010.

J.R. Hobbs. Resolving pronoun references. *Lingua*, 44(4):311–338, 1978.

J. Hoffart, F.M. Suchanek, K. Berberich, and G. Weikum. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 2012.

S. Huang, Y. Zhang, J. Zhou, and J. Chen. Coreference resolution using Markov Logic Networks. In *Proceedings of CICLing*, pages 157–168, 2009.

R. Iida, K. Inui, H. Takamura, and Y. Matsumoto. Incorporating contextual cues in trainable models for coreference resolution. In *Proceedings of the 10th EACL Workshop on the Computational Treatment of Anaphora*, pages 23–30. Citeseer, 2003.

R. Isele, J. Umbrich, C. Bizer, and Andreas Harth. LDSpider: An open-source crawling framework for the web of linked data. In *Proceedings of 9th International Semantic Web Conference (ISWC 2010) Posters and Demos*, 2010.

H. Ji, R. Grishman, H.T. Dang, K. Griffitt, and J. Ellis. Overview of the tac 2010 knowledge base population track. In *Proceedings of the Third Text Analysis Conference*, 2010.

H. Ji, R. Grishman, and H. T. Dang. Overview of the tac 2011 summarization track: Guided task and aesop task. In *Proceedings of the Text Analysis Conference (TAC 2011), Gaithersburg, Maryland, USA, November*, 2011.

R. Krovetz. More than one sense per discourse. In *NEC Princeton NJ Labs., Research Memorandum*, 1998.

S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9.

H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34. Association for Computational Linguistics, 2011.

D. B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

J. Li, Z. Zhang, X. Li, and H. Chen. Kernel-based learning for biomedical relation extraction. *Journal of the American Society for Information Science and Technology*, 59(5):756–769, 2008.

C. Matuszek, J. Cabral, M. Witbrock, and J. DeOliveira. An introduction to the syntax and content of cyc. In *Proceedings of the 2006 AAAI Spring*

Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, volume 3864, pages 44–49. AAAI Press, 2006.

P. McNamee and H.T. Dang. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, 2009.

O. Medelyan and C. Legg. Integrating cyc and wikipedia: Folksonomy meets rigorously defined common-sense. In *Proceedings of the WIKI-AI: Wikipedia and AI Workshop at the AAAI08 Conference, Chicago, US*, 2008.

P.N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.

R. Mihalcea. Using wikipedia for automatic word sense disambiguation. In *Proceedings of NAACL HLT*, volume 2007, pages 196–203. Association for Computational Linguistics, 2007.

R. Mihalcea and P. Edmonds, editors. *Proceedings of SENSEVAL-3*, Barcelona, Spain, July 2004.

D. Milne and I. H. Witten. Learning to link with Wikipedia. In *CIKM '08: Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518, NY, USA, 2008. ACM.

D. Milne and I.H. Witten. An open-source toolkit for mining wikipedia. In *Proc. New Zealand Computer Science Research Student Conf., NZCSRSC*, volume 9, 2009.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the ACL-IJCNLP 2009*, pages 1003–1011. ACL, 2009.

R. Navigli and S. P. Ponzetto. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* Uppsala, Sweden, 11–16 July 2010, pages 216–225, 2010.

R. Navigli and S.P. Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

R. Navigli, K.C. Litkowski, and O. Hargraves. Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 30–35. Association for Computational Linguistics, 2007.

M. Negri and M. Kouylekov. Fbk_nk: A wordnet-based system for multiway classification of semantic relations. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 202–205. Association for Computational Linguistics, 2010.

V. Ng. Learning noun phrase anaphoricity to improve coreference resolution: issues in representation and optimization. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 151–158, 2004.

V. Ng. Semantic class induction and coreference resolution. In *Proceedings of the ACL*, volume 45, pages 536–543, 2007.

V. Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

V. Ng and C. Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 104–111, 2002.

T.V.T. Nguyen and A. Moschitti. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of ACL-HLT*, pages 277–282. ACL, 2011.

T.V.T. Nguyen, A. Moschitti, and G. Riccardi. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1378–1387. Association for Computational Linguistics, 2009.

E.W. Noreen. *Computer-Intensive Methods for Testing Hypotheses : An Introduction*. Wiley-Interscience, April 1989. ISBN 0471611360.

T. Pedersen. A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation. *NAACL 2000*, pages 63–69, 2000.

T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics, 2004.

E. Pianta, C. Girardi, and R. Zanoli. The textpro tool suite. In *Proceedings of LREC*, volume 8, 2008.

M. Poesio, R. Mehta, A. Maroudas, and J. Hitzeman. Learning to resolve bridging references. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 143–150. Association for Computational Linguistics, 2004.

S. P. Ponzetto and M. Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 192–199, 2006.

S.P. Ponzetto and R. Navigli. Knowledge-rich word sense disambiguation rivaling supervised systems. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1522–1531. Association for Computational Linguistics, 2010.

H. Poon and P. Domingos. Joint inference in information extraction. In *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pages 913–918, 2007.

H. Poon and P. Domingos. Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 650–659, 2008.

S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky. Shallow semantic parsing using support vector machines. In *Proceedings of HLT/NAACL*, pages 2–7, 2004.

S. Pradhan, E. Loper, D. Dligach, and M. Palmer. Semeval-2007 Task-17: English Lexical Sample, SRL and All Words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in*

Natural Language Processing, pages 492–501. Association for Computational Linguistics, 2010.

A. Rahman and V. Ng. Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 814–824. Association for Computational Linguistics, 2011.

C. Ramakrishnan, K. Kochut, and A. Sheth. A framework for schema-driven relationship discovery from unstructured text. In *The Semantic Web-ISWC 2006*, pages 583–596. Springer, 2006.

L. Ratinov and D. Roth. Learning-based multi-sieve co-reference resolution with knowledge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, (EMNLP-CoNLL 2012)*, pages 1234–1244, Jeju Island, Korea, 2012.

L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 1375–1384, 2011.

S. Riedel and I. Meza-Ruiz. Collective semantic role labelling with markov logic. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 193–197, 2008.

B. Rink and S. Harabagiu. Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 256–259. Association for Computational Linguistics, 2010.

B. Rosario and M.A. Hearst. Classifying semantic relations in bioscience texts. In *ACL 2004*, pages 430–437. ACL, 2004.

D. Roth and W. Yih. Probabilistic reasoning for entity & relation recognition. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

S. Sahay, S. Mukherjea, E. Agichtein, E. V. Garcia, S. B. Navathe, and A. Ram. Discovering semantic biomedical relations utilizing the web. *ACM Trans. Knowl. Discov. Data*, 2:3:1–3:15, April 2008. ISSN 1556-4681.

S. Sarjant, C. Legg, M. Robinson, and O. Medelyan. All you can eat ontology-building: feeding wikipedia to cyc. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 341–348. IEEE Computer Society, 2009.

K.K. Schuler. Verbnet: A broad-coverage, comprehensive verb lexicon. *Dissertations available from ProQuest*, 2005. URL `http://repository.upenn.edu/dissertations/AAI3179808`.

F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47, March 2002.

N. Shadbolt, W. Hall, and T. Berners-Lee. The semantic web revisited. *Intelligent Systems, IEEE*, 21(3):96–101, 2006.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach

to coreference resolution of noun phrases. *Computational Linguistic*, 27 (4):521–544, 2001.

F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 697–706, New York, NY, USA, 2007. ACM Press. ISBN 9781595936547.

G. Szarvas and I. Gurevych. Tud: semantic relatedness for relation classification. In *Proceedings of SemEval-2010*, pages 210–213. ACL, 2010.

S. Tratz and E. Hovy. Isi: Automatic classification of relations between nominals using a maximum entropy classifier. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 222–225. Association for Computational Linguistics, 2010.

K. Tymoshenko and C. Giuliano. Fbk-irst: Semantic relation extraction using cyc. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 214–217, 2010.

M. Van Assem, A. Gangemi, and G. Schreiber. Conversion of wordnet to a standard rdf/owl representation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC06), Genoa, Italy*, pages 237 – 242, 2006.

S. Van Landeghem, T. Abeel, Y. Saeys, and Y. Van de Peer. Discriminative and informative features for biomolecular text mining with ensemble feature selection. *Bioinformatics*, 26(18):i554–i560, 2010.

V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998. ISBN 0471030031.

Y. Versley, S.P. Ponzetto, M. Poesio, V. Eidelman, A. Jern, J. Smith, X. Yang, and A. Moschitti. Bart: A modular toolkit for coreference

resolution. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session*, pages 9–12. Association for Computational Linguistics, 2008.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. A model-theoretic coreference scoring scheme. In *MUC6 '95: Proceedings of the 6th conference on Message understanding*, pages 45–52, 1995.

Y. Wilks. An intelligent analyzer and understander of english. *Communications of the ACM*, 18(5):264–274, 1975.

T. Winograd. Understanding natural language. *Cognitive Psychology*, 3 (1):1–191, 1972.

E. Wolf, UKP TU, and I. Gurevych. Aligning Sense Inventories in Wikipedia and WordNet. In *Proceedings of the 1st Workshop on Automated Knowledge Base Construction*, pages 24–28, 2010.

X. Yang and J. Su. Coreference resolution using semantic relatedness information from automatically discovered patterns. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 528–535, June 2007.

X. Yang, G. Zhou, J. Su, and C.L. Tan. Coreference resolution using competition learning approach. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 176–183. Association for Computational Linguistics, 2003.

L. Yao, S. Riedel, and A. McCallum. Collective cross-document relation extraction without labelled data. In *Proceedings of EMNLP 2010*, pages 1013–1023. ACL, 2010.

D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association*

for Computational Linguistics, pages 189–196. Association for Computational Linguistics, 1995.

A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland. Textrunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 25–26. Association for Computational Linguistics, 2007.

D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106, 2003.

T. Zesch, C. Müller, and I. Gurevych. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, 2008.

GD Zhou, J. Su, J. Zhang, and M. Zhang. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 1456–1466, 2005.

Y. Zhou, L. Nie, O. Rouhani-Kalleh, F. Vasile, and S. Gaffney. Resolving surface forms to wikipedia topics. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1335–1343. Association for Computational Linguistics, 2010.