Natural Language Processing for Semiautomatic Semantics Extraction: Encyclopedic Entry Disambiguation and Relationship Extraction using Wikipedia and WordNet

Dissertation written by: Maria Ruiz-Casado.

Under supervision of: Pablo Castells.

Escuela Politécnica Superior.

Universidad Autónoma de Madrid.

September 8, 2009

To the memory of my father.

Contents

List of Figures

List of Tables

Abstract

The World Wide Web includes an extremely large quantity of information, with contents presented mainly in the form of natural language, whose ambiguities are hard to be processed by a machine. Within the current trend of Semantic Technologies, several efforts try to address this problem by learning structured knowledge from unrestricted text, for example by means of Information Extraction techniques using Natural Language Processing (NLP). Annotating natural language against a structured knowledge base, e.g. an ontology, eases the automatic processing of web contents.

This work explores NLP techniques for the automatic extraction of semantics from non-structured text. Two tasks are on main focus: word sense disambiguation and relations extraction.

The methods proposed include a new semi-automatic approach to the acquisition and generalization of lexical patterns used for the extraction of semantic relations, a new pruning methodology to select patterns based on rote extractors, and a method to disambiguate encyclopaedic entries against WordNet senses.

A prototype has been implemented to test the above-mentioned methods, extracting knowledge from Wikipedia using the World Wide Web and WordNet as data sources.

Resumen

La Web actual incluye una enorme cantidad de información, con contenidos presentados principalmente en forma de lenguaje natural, cuyas ambigüedades son difíciles de procesar por una máquina. Las tendencias actuales en Tecnologías Semánticas presentan diversos esfuerzos para resolver este problema adquiriendo conocimiento estructurado de texto libre, por ejemplo usando técnicas de Extracción de Información mediante Procesamiento de Lenguaje Natural (PLN). Una aproximación común es la anotación de lenguaje natural según una base de conocimiento estructurada, por ejemplo una ontología, lo cual facilita el procesamiento automático de contenidos Web.

Este trabajo explora técnicas de Procesamiento de Lenguaje Natural (PLN) para la extracción automática de semántica a partir de texto no estructurado. Dos tareas están en el punto de mira: desambiguación de sentidos y extracción de relaciones.

Los métodos propuestos incluyen una nueva aproximación semi-automática para adquirir y generalizar patrones léxicos usados en extracción de relaciones semánticas, una nueva metodología de filtrado en la selección de patrones basada en extractores de repeticiones, y un método para desambiguar artículos enciclopédicos según los sentidos de WordNet.

Se ha implementado un prototipo para probar los métodos mencionados, extrayendo conocimiento de Wikipedia usando la Web y WordNet como fuentes de datos.

Acknowledgements

This thesis work started in the year 2003. There have been many people to whom I am indebted for their implication in the shape of technical and professional help, fundings, time and encouragement.

I wish to thank the Universidad Autonoma de Madrid and the Higher Polytechnical School for making possible this thesis. Thanks Pablo, Manuel, Diana, Fran, Leila, Pablo, Estefanía, Pedro, Miguel, Manu, Jordi, Rosa, Álvaro, Germán, Mariano, Ismael, Pilar, Roberto, Estrella, Mick, Alfonso, Marina and many other who helped, gave ideas and with whom I shared good moments. To this later point I must mention family, very specially the two Mariangel and the cuberada in general.

I wish to thank the Tokyo Institute of Technology and the Okumura Lab in the Precision and Intelligence Laboratory, for making possible a one-year research stay in Japan that is part of this thesis work. Thanks to all the lab for their help and kindness, and very specially to Okumura-sensei, Takamura-san, Sugiyama-san, Yoshida-san and Kim-san. Also to Ines, Yuri and Bea for the good times there.

Thanks to the Dow Chemicals' colleagues in Switzerland. Although it was not possible to apply to my thesis the works carried out there, I am indebted for the time and encouragement that they offered: Erik, Michael, Steve, Theo, Anca and other colleagues.

Thanks Kisko, Antonio, Virginia, Laia and Pol. To those who were always there: Mar, Imma, Marc, Ana, Sonia, Juan, Julia, David.

I owe this thesis specially to you, with all my love: Enrique, Irene and Angeles.

Agradecimientos

Esta tesis arrancó en el año 2003 y han sido muchas las personas a las que debo agradecer por su implicación en forma de ayuda técnica, profesional, financiación o motivación durante estos años.

Deseo agradecer a la Universidad Autónoma de Madrid y a la Escuela Politécnica Superior el haber hecho posible esta tesis doctoral. Gracias Pablo, Manuel, Diana, Fran, Leila, Pablo, Estefanía, Pedro, Miguel, Manu, Jordi, Rosa, Álvaro, Germán, Mariano, Ismael, Pilar, Roberto, Estrella, Mick, Alfonso, Marina y tantas otras personas que en algún momento me ayudaron, dieron ideas y/o con las que compartí ratos agradables. A razón de este último punto, gracias también a familia y allegados, muy especialmente a las Mariangel y la cuberada en general.

Deseo agradecer al Tokyo Institute of Technology y al Okumura Lab del Precision and Intelligence Laboratory por haber hecho posible un año de estancia investigadora en Japón que forma parte de esta tesis. Gracias a todo el laboratorio por la ayuda y la cordialidad recibida. Muy en particular quiero agradecer a Okumura-sensei, Takamura-san, Sugiyama-san, Yoshida-san, Kim-san. También a Inés, Yuri y Bea, por los buenos momentos compartidos allí.

Gracias a los compañeros de Dow Chemicals en Suiza. Aunque no pude aplicar a mi tesis los trabajos que allí desarrollé, quiero agradecer el tiempo y la motivación que me ofrecieron: Erik, Michael, Steve, Theo, Anca y demás compañeros.

Gracias a Kisko, Antonio, Virginia, Laia y Pol. A los que siempre habéis estado allí: Mar, Imma, Marc, Ana, Sonia, Juan, Julia, David.

Esta tesis os la debo especialmente a vosotros, con todo mi amor: Enrique, Irene y Ángeles.

Chapter 1

Introduction

1.1 Semantic Technologies

Natural Language Processing (NLP) is a well established area between Artificial Intelligence and Linguistics. As early as the fifties in the past century, computational linguists started addressing problems in the automatic processing of textual data for Machine Translation. In the sixties, the term *Computational Linguistics* was coined, and research continued on topics such as syntactic analysis and word sense disambiguation.

These early works were limited by the restricted computational capability and the still reduced availability of textual data in electronic format. The current scope in the area (in terms of technology and available resources) is quite different, and after almost six decades of research in natural language processing the technology has evolved considerably, but still many of the problems on automatic processing identified in the early times are far from being fully resolved yet.

Since the very beginning the applications of NLP have not ceased growing, including areas such as Machine Translation, Information Extraction, Information Retrieval, Automatic Text Summarization or Sentiment Analysis, among others. In particular, Information Extraction deals with the problem of extracting meaningful structured data from unstructured data such as natural language texts. The topic of this work falls within the scope of the Information Extraction field.

From the nineties, the World Wide Web represented a revolution in the availability of electronic textual content. Since the first World Wide Web project presented by Tim Berners Lee in 1989, the advances in web technology have been remarkable. These advances include the development of the HTML language and the HTTP protocol (which eased the creation and transfer of contents integrating text and hypermedia), the dynamic generation of web pages, the development of web browsers and search engines and the enhancement of the user interaction. These gave way to an overwhelming availability of text in electronic format. The web size is estimated today at more than 25×10^9 indexable pages[1]. The *deep web*, formed by the information that grounds the dynamically generated web pages, stored in underlying data bases, is estimated to be greater than

[1] http://worldwidewebsize.com as of February 2009. Sorted on Google, Yahoo, Windows Live Search and Ask.

the total volume of printed information existing in the world [Castells, 2003].

The availability of such a big quantity of contents has converted the World Wide Web into a universal information resource, as almost each and all general information can be found in it, although not always the quality of the available data satisfies the user's expectations. The data spectrum in terms of language quality and contents structure varies greatly: from free on-line knowledge resources, sometimes collaboratively built and very well structured like Wikipedia, which competes in quality with some well known encyclopedias developed by professionals, to blogs, forums and community networks where the information is looser and the contents are subjective, sometimes very inexact and the language in use is colloquial or even slang. The huge quantity of contents and the mix of information sources with diverse qualities cause that the results returned by the search engines often include irrelevant information along with the desired data.

Berners-Lee et al. [2001] and Fensel et al. [2002] report many examples where the exponential growth of the World Wide Web and its huge availability of data make the tasks of searching, retrieving data, and maintaining contents a hard and time consuming task when these tasks (or part of them) have to be carried out manually. By the end of the past decade the vision of the *Semantic Web* (SW) emerged. Since Berners Lee introduced this vision, an important research community in universities, governments and private enterprises joined efforts in the search of a web where the content would be provided with an underlying structure, organised in such a way that information irrelevant to the desired task (search and retrieval, content management, etc.) could be discarded and only relevant information would be selected, allowing an automation of the information processing and sharing: automated procedures or agents would negotiate to share information and would come up with *only* the right and best suited answer(s) to a query when searching; they would manage web content with minimal human supervision, also maintaining and updating information in an error-free way. Although some steps have been taken in that direction, the task is complex, being one of the recognised difficulties that prevents the complete automation of those processes the fact that the contents on the Internet are presented mainly in natural language, whose ambiguities are hard to be processed by a machine [Ding et al., 2002].

The natural language used by humans is ambiguous. All human languages involve a certain degree of polysemy in their vocabularies. This means that a single lexical string can be used to represent more than one single concept, for instance, the word *bank* in English has twelve different meanings in the on-line Merriam-Webster dictionary:

- A mound, pile or ridge raised above the surrounding level.
- The rising ground bordering a lake, river or sea (...).
- A steep slope (...).
- The lateral inward tilt of a surface along a curve or a vehicle (as an air plane) when taking a curve.
- A protective or cushioning rim or piece.
- An establishment for the custody, loan or exchange, or issue of money (...)
- The table, counter or place of business of a money changer.
- A person conducting a gambling house or game

- A supply of something held in reserve (...)
- A place where something is held available (...)
- A group of series or objects arranged together in a row or a tier.
- One of the horizontal and usually secondary of lower division of a headline.

Some words are more polysemous than others, for instance, the noun *hand* has in the WordNet lexicon fourteen senses, while the noun *nail* has only three senses. Some words are monosemous in a dictionary, for instance, *Indian elephant* has one single sense in the above-mentioned lexicon. Some semanticists point out that highly polysemous words represent very common, general concepts while more rare and specific concepts tend to be represented by monosemous words [Magnus, 2001]. In any case, polysemy in natural language causes that, when retrieving information with the purpose of answering a certain query, the results returned by a search engine usually mix more than one of the possible senses of the word.

In order to avoid word sense ambiguities and make explicit the meaning underlying the data, and therefore processable by a machine, a common practise is the annotation of the sense of certain words, pages or other web resources using a sense repository, which can be, for example, represented as a knowledge representation formalism called *ontology*. This formalism, explained in detail in Chapter 2, contains the words that represent a certain domain, including the different senses of polysemous words, and relationships amongst them. Some of the most common relationships are the subclass relations or the part-of relations. For instance, if we have the concepts *dog* and *mammal*, we can say that the first one is a subclass of the second one. Equally, there is a part-of relationship between *tail* and *dog*. Through their different relationships, the senses of a polysemous words can be distinguished. Following the above example of the word *bank*, the *riverside* meaning can be distinguished from the *financial institution* sense in the ontology. Although they are concepts that share the same lexical string, the first sense will be a subclass of the concept *geological formation*, and the last one will be a subclass of *institution, organisation*. The annotation in the web page of the word *bank* will point to one of its possible meanings as represented in the ontology.

Today the vision of the Semantic Web is wider and softer, evolving to the so called Semantic-based Technologies. The complexity of producing a web supported by a fully structured content organisation is acknowledged, and rather than targeting a fully formal Semantic Web that would allow a full automatising, the research community seeks continuous improvements on the tasks of searching, retrieving data, extracting content and annotating, among other, through the use of semantic techniques. These techniques were founded in many cases in those initial efforts during the last century, and are nowadays developed from diverse communities joining ideas from the Semantic Web, Computational Linguistics and Machine Learning, where the common goal is to provide an incremental advance in the semantic enrichment of data that makes our current web easier to process. The presence of semantic technologies to aid knowledge management is spreading widely, and is already in use in our today web, e.g. through the main search engines such as Yahoo, Google or Bing, which also offer a limited processing of natural language queries.

1.2 Motivation and Goals

One of the applications of the semantic technologies is annotating hypermedia contents with semantic information. Under this view, the relevant concepts appearing in web pages should be annotated with the identifier of a concept inside an ontology, is such a way that the plain text is bridged to structured knowledge. The assignment of such tags receives the name of *semantic annotation*. Through the annotation of the web contents, the meaning of salient words in a web page is explicitly marked, thus making the annotated text automatically processable by a machine without meaning ambiguities.

In practise, manually building and maintaining ontologies is a very time and resource-consuming task, as is annotating web pages. This difficulty is a long-time known bottleneck for the goal of enriching the web with semantic content, and many authors have pointed out the need for automatic or semi-automatic procedures to fill the gap between unstructured text and structured knowledge [Contreras et al., 2003, Contreras, 2004, Gómez-Pérez et al., 2003, Kietz et al., 2000, Maedche and Volz, 2001, Maedche and Staab, 2000].

Addressing the problem of enriching hypermedia text by means of structured annotation, this work focuses on the semi-automation of knowledge identification for the enrichment of ontologies and semantic annotation. Amongst the many possible approaches, this work is centered in the application and state-of-the-art improvement of Natural Language Processing (NLP) techniques.

The main goal of this study is the design and implementation of NLP techniques to help in annotating texts and extending ontologies. The following tasks are addressed:

1. Improve on automatic Word-Sense Disambiguation techniques for identifying the sense with which a word is used inside a context. For instance, if we have the following sentences,

   (1) After bathing in the lake, I took a sunbath in the bank.

   (2) The loan I managed to get in that bank is very convenient.

   we can guess the meaning of *bank* in each of them using the evidence conveyed by the words that surround it (e.g. *lake*, in the first sentence, and *loan*, in the second one).

2. Improve on automatic extension of ontologies by learning new relationships between the concepts from free text. For example, the following sentence,

   (3) The happy dog wagged its tail.

   indicates that *tail* is a part of *dog*, because of the possessive pronoun *its*.

To tackle these problems several methods are proposed here, mainly based on (1) the use of the WordNet lexicon as external lexico-semantic source of information, (2) statistical methods using the Vector Space Model for the disambiguation task, (3) pattern extraction and automatic generalisation of them for the detection of relationships between terms, (4) rote extractor-based methods to categorize and prune relationships.

These methods have been tested to check their ability in performing the intended goals. An experiment using the controlled vocabulary of an on-line encyclopedia, the Wikipedia, has been carried out, disambiguating the encyclopedia entries as per the word senses in the lexicon and enriching the lexicon with the encyclopedic definitions and with new relations found in the definition texts.

1.3 Structure of this Document

This document is structured as follows:

- Chapter 2 provides an introduction to to ontology design and construction, and text annotation. It ends with a brief introduction to the different analysis levels of Natural Language Processing.
- Chapter 3 revises the main approaches to semantic annotation using Natural Language Processing. Techniques on Word Sense Disambiguation and Information Extraction are reviewed, including past and current relevant works, international competitions and up-to-date results on these topics. Some research opportunities are also discussed.
- Chapter 4 describes the general architecture of a system designed for this work, which uses an approach based on Natural Language Processing to disambiguate and capture semantic relations from texts. This chapter describes the techniques used, their implementation, tests results and evaluation.
- Finally, Chapter 5 draws some conclusions and open lines for future research.

Chapter 2

Preliminaries

2.1 Ontologies

An ontology is a formalism for knowledge representation. Ontologies provide a common vocabulary of an area and define the meaning of terms (*concepts*) and the relations between them [Gómez-Pérez et al., 2003]. One of the most referred definitions in the literature is that given by Gruber [Gruber, 1993], which states that an ontology is a "formal and explicit representation of a conceptualisation for a domain". Let us look inside this definition:

- Formal and explicit. This means that the representation of the data meaning is given in a formal way, following certain specifications, making explicit in the ontology the meaning in order to make it processable by a machine.
- Conceptualisation for a domain. The domain is modelled via the concepts that are included in the ontology. The concepts included in the domain, and how they are ordered and related inside the ontology reduce or eliminate conceptual and terminological confusion when sharing information.

Figure 2.1 shows an example of how the domain concepts are represented inside an ontology. The ontology is formed by a *class hierarchy*, which has the form of an inverted tree of classes, where each class represents a concept, and the concepts are ordered through a *subsumption relation*, that is, in such a way that, departing from a general root, the more general concepts are higher in the hierarchy and have a meaning that subsumes the concepts below them. The subsumption relation is also called *is-a*, or *is-a-type-of* relation, so the concepts located right below a higher concept, hanging of it as *sons*, are *hyponyms* of the *father* concept, which is also called *hyperonym* or *hypernym*. The *is-a-type-of* relation is therefore also called *hyponymy*, as the relation between the sons and the father concept, and accordingly the *has-the-type* relation is also called *hypernymy*, as the relation that the hyperonym maintains with its hyponyms. The hyponymy relation is kept throughout the entire hierarchy in a way that, the lower the position of a concept in the hierarchy, the most specific is its meaning (and vice-versa: the higher concepts have a more

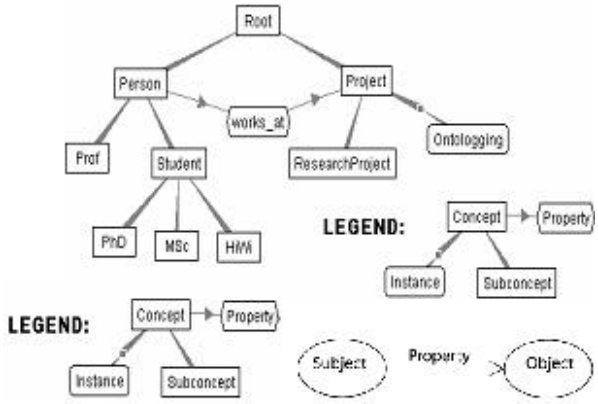

Figure 2.1: Sample ontology fragment, from Maedche and Volz [2001].

general meaning). To finalise with the terminology related with this relation, a hierarchy based on the subsumption relation receives also the name of *taxonomy*. The example in Figure 2.1 presents a very small hierarchy composed only by the root node, that subsumes the concepts "Person" and "Project". This domain represents only these two types of concepts, and could be used, for instance, to represent in a very simple way a web page of projects in a university department. Two types of Person are considered: "Professor" and "Student". One step further, the concept "Student" subsumes the various typology of students: "PhD", "MSc" and "HiWi". The branch that departs from the concept "Project" goes to a more specific child: "Research project".

The concepts in the hierarchy are further related through *non-taxonomic relationships*. Following with the example in Figure 2.1, the relation WorksAt links the concept Person and Project, to explicitly mark that a Person in that domain can work in a project. The relations consist in *triples*, including a *subject*, an *object* and a *property* that defines the semantic relation between them. Sometimes the relation is not established between two concepts in the hierarchy: this happens when the property defined is a *feature* or a *restriction* in the cardinality or type that the object can take. In that case the triple does not point to another concept, but to the corresponding cardinal or type value. For example, the concept "Person" could have a relation defined by the property "age" pointing to an integer value, thus indicating that there is a restriction: the object has to be an integer value representing the age of the person. These properties are called *attributes*. The ontology is built by linking pairwise those related concepts in the hierarchy through the semantic relations. The lower concepts in the hierarchy can *inherit* the relations from their ancestors. In the example, the relation *Person WorksAt Project* also holds for *Professor WorksAt Project*. Finally, the concepts in the ontology are *instantiated*. Each class or concept covers the meaning of a set of *instances* that follow all the properties that typify the class or concept to which they pertain. For example, the instances of the concept "student" could be "John Brown", "Peter Smith", etc. and the instances of the concept "project" would be "Ontologging", etc. The instances have to be completed with the value of their attributes, as well, e.g., if the attribute "age" is defined, "John

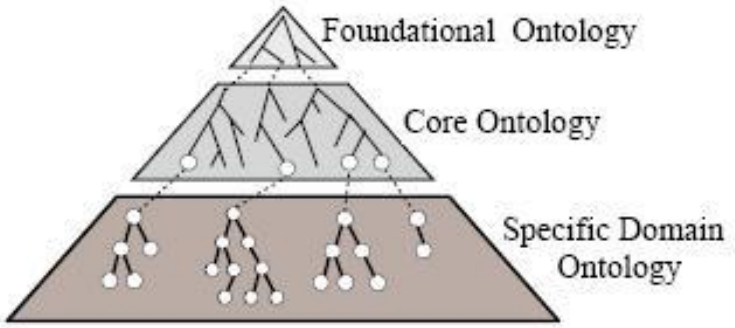

Figure 2.2: Level of generality in an ontology, from Navigli and Velardi [2003].

Brown" would point to "23" through this attribute, "Peter Smith" to "22", etc. Above this model of knowledge, the ontologies for the Semantic Web include logical rules which, by means of predicate calculus, allow the inference of new meaning. For instance, it is possible to define new classes from those that match the rule of having a particular relationship.

Navigli and Velardi [2003] distinguish three levels of generality in a domain ontology, as shown in Figure 2.2. The most general levels of the ontology consist in basic kinds of concepts considered to be domain independent. These topmost categories of entities and their relations conform the *Foundational Ontology*, generally domain independent to ensure re-usability across different domains. Deeper in the representation of knowledge, the *Core Ontology* usually includes a few hundred application-domain concepts. A core ontology is generally used for general texts like news, etc, where a fine grained distinction between word senses is not needed. Specific domain texts will require a *Specific Domain Ontology*, covering as many domain-specific concepts and as many sense distinctions as required by the members of the user community for that domain.

Once an ontology has been selected, it is possible to annotate the words in a text to indicate which is the concept represented by each of them. Figure 2.3 shows a web page fragment with the semantic annotations to the concepts in an ontology.

Ontologies are used as knowledge resource in different disciplines. They are employed, for instance, in the field of Natural Language Generation [Bateman, 2002] as a source of knowledge that, combined with grammars and other resources make possible the automatic generation of texts written in natural language. As reported by Gómez-Pérez et al. [2002], some other applications of ontologies include Knowledge Management (On-To-Knowledge[1], CoMMA[2], Onto Web[3], etc), e-commerce (OBELIX[4]), Natural Language Processing (OntoTerm[5]), Automated Translation (MikroKosmos[6]), and Information Retrieval (SemanticMiner[7]).

[1] http://www.ontoknowledge.org
[2] http://www.ercim.org/publication/Ercim_News/enw41/dieng.html
[3] http://www.ontoweb.org
[4] http://www.cs.vu.nl/%7Ezykov/imse/projects/projects.html
[5] http://www.ontoterm.com/
[6] http://crl.nmsu.edu/users/sb/papers/thai/node1.html
[7] http://www.ontoprise.de/content/e3/e24/index_eng.html

Figure 2.3: Semantic annotation using an ontology, from Popov et al. [2003].

In the Semantic Web context, ontologies are not only used to process contents for the retrieval of documents or for the extraction of information, but also as a support to the interaction and performance of web services. The ontologies allow the description of the functionalities and procedures of the web services: inputs, outputs, functional restrictions, procedural steps when interacting with other services, etc. As far as the operation of a given service is made explicit in the ontology, a machine can process and understand how different services behave, in such a way that it is possible to search the services required for a certain task, compose them and execute the task, all done in an automatic way.

In 2001, Berners Lee [Berners-Lee et al., 2001] pictured a world wide web where ubiquitous semantic agents and semantic search engines could automatise negotiation, information exchange and retrieval, extending current capabilities and eliminating the barriers of handling very large, natural language, on-line media. This picture is so far incomplete. Although during the present decade there has been an incremental availability of semantically annotated web resources following ontologies that can be found in some ontology libraries (WebOnto[8], Ontolingua[9], DAML Ontology Library System[10], SHOE[11], Ontology Server[12], IEEE Standard Upper Ontology[13], SESAME[14], OntoServer[15]), there are still many steps to take. There is a great effort being done by the research community in the specification and enhancement of web services for the Semantic Web [Hendler, 2001]; also there is a continuous development of tools based on the Semantic Web languages. Other open problems deal with the integration of different ontologies that represent a same or similar domains [Calvanese et al., 2002], or how to adapt ontologies to

[8] http://eldora.open.ac.uk:3000/webonto
[9] http://www-ksl-svc.stanford.edu:5915/
[10] http://www.daml.org/ontologies/
[11] http://cs.umd.edu/projects/plus/SHOE/
[12] http://starlab.vub.ac.be/research/dogma/OntologyServer.htm
[13] http://suo.ieee.org/
[14] http://www.openrdf.org/
[15] http://www.aifb.uni-karlsruhe.de/Projekte/viewProjektenglish?id_db=24

rapid evolving domains and how to control ontology evolution [Klein and Fensel, 2002].

In the following sections we will focus on works regarding ontology acquisition and text annotation. Section 2.2 focuses in the problem of ontology building and the most common tools for the manual development of ontological knowledge. Section 2.3 introduces some approaches to the semi-automation of ontology acquisition, Section 2.4 reviews semi-automatic web annotation and Section 2.5 outlines the basic Natural Language Processing levels and which depth of analysis is required for the ontology enrichment and annotation purposes.

2.2 Ontology languages and tools

The Semantic Web community has developed a series of languages that allow the representation of ontologies and the annotation of web pages: RDF[16], DAML+OIL[17], OWL[18], amongst other. They make possible the use of tags to mark the concepts in a (hyper-)text, in order to refer them to a pre-defined structure where the knowledge if represented.

XML allows the definition of ordered structures by means of tagged concepts, where attributes can be defined for each tag, but is not appropriate to define an ontology because it does not provide markup mechanisms for defining a class hierarchy, distinguishing concepts, instances and relations, and it does not provide inheritance of properties. Due to this fact, although facilitating the use of tags to mark concepts, XML is not always considered a language for the Semantic Web.

RDF appears in 1999 as a new language specifically designed to define ontologies. It permits the definition of a class hierarchy, discriminates classes (concepts) and instances, allows the definition of relations in the form of triples *subject-property-object*, amongst other features. RDF was followed by DAML+OIL, a language that arose from the union of two similar languages (DAML and OIL) developed as a enhancement of RDF. OWL appears also as an evolution of RDF, including all its features and extending them. For instance, OWL allows reasoning through the use of logical expressions and the definition of properties over the relations (symmetry, transitivity, etc).

There are several tools that use these languages for the annotation of hypertext using a background ontology: COHSE[19], OntoAnnotate[20], OntoMatAnnotizer[21], amongst other. Some are editors that aid the task of manually annotating pages, assisting the ontologist with interfaces to select words from the text, to select concepts from the background ontology(ies), and linking them by means of an URI[22]. Some of them incorporate tools to semi-automatise part of the tasks. In the following sections some of the semi-automated solutions are presented.

Regarding the construction of the ontologies themselves, there are also several tools developed to assist the ontologist in creating, editing, storing and managing them. Directly writing languages like RDF or OWL is difficult and usually error prone. The most well known graphic frameworks

[16] http://www.w3.org/RDF/
[17] http://www.w3.org/TR/daml+oil-reference/
[18] http://www.w3.org/2001/sw/WebOnt/
[19] http://cohse.semanticweb.org/
[20] http://www.ontoprise.de/content/index_eng.html
[21] http://annotation.semanticweb.org/tools/
[22] A Universal Resource Identifier is the common artifact used to annotate words with concepts inside an ontology

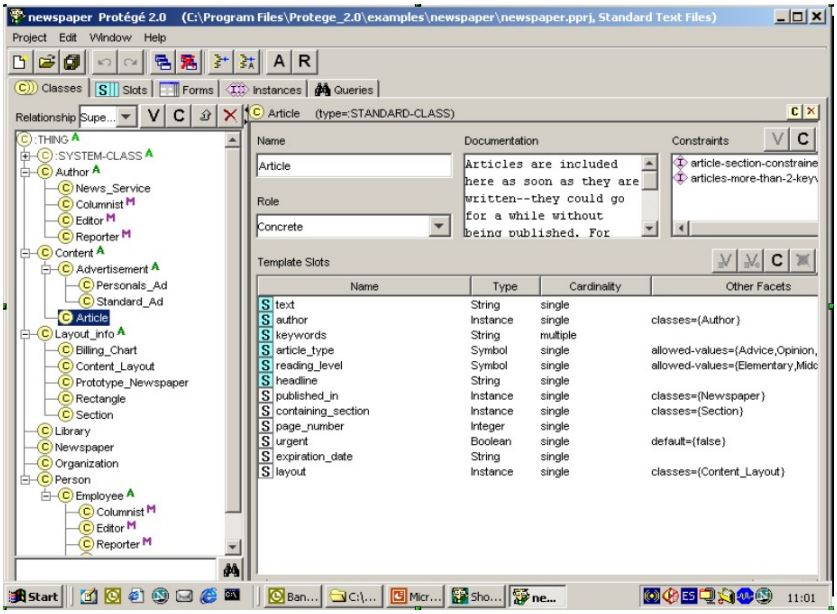

Figure 2.4: Screenshot of the Protégé system while editing an ontology.

for the management of ontologies are Protégé[23] and Sesame[24].

Figure 2.4 shows a screen capture from the editor Protégé. Part of an ontology for the newspaper domain can be observed. In order to manually create an ontology this tool requires at first the definition of the concept hierarchy. Then, for each concept, where appropriate, the relationships with other concepts and attributes have to be defined. To scale down this task, specially for domains where a large number of concepts converge, a common practise to create a new ontology is to re-use existing hierarchies, specially when there exist categorised vocabularies or thesauri previously exposed to consensus in the domain. These hierarchies can be found for several disciplines, like medicine (MeSH[25]), human resources (HR-XML[26]), etc. Some of them can be found in XML format and can be easily ported to RDF, to be completed by hand.

Once the structure of the ontology is defined with its concepts, subconcepts and relationships, it is necessary to *populate* it: the instances for each concept have to be entered. This means that those words that are expected to appear in our domain texts have to be linked to the concepts that represent them. In Figure 2.4 we can see that the concept *Article* is selected in the hierarchy, at the left of the screen, and the right part of the screen shows the properties to be filled for each instance of *Article*: its text, author, headline, the Newspaper where it is published, etc. Each defined relation and attribute receives its corresponding value. This work has to be completed for each instance of *Article*, that is, for each newspaper article in the data base of the news ontology. The work populating with instances has to be done for the rest of the ontology classes in the

[23] http://protege.semanticweb.org
[24] http://sesame.aidministrator.nl/
[25] http://www.nlm.nih.gov/mesh/
[26] http://www.hr-xml.org

domain of a newspaper: other texts like advertisement texts, data about people, as columnists (name, position, relations with the articles they wrote, etc) or editors, and sometimes the ontology is required to include data about the newspaper structure or about the organisation that publish it.

Populating an ontology can result in a very high cost when working in a manual way. The example ontology for news introduced above illustrates a case where creating a new ontology can be very hard for domains of broad knowledge coverage. Nevertheless, this complexity can also be found in very specialised domains that require a very fine-grained ontology, with large vocabularies of concepts and several relationships between them. Furthermore, once the ontologies are created, the texts have to be annotated using them, adding an extra effort when annotating in a manual way. All these tasks, required to get semantically annotated contents, become unfeasible when the goal is to adapt high amounts of preexisting, unannotated contents, even when they are partly ordered and some of the required information is stored in a data base, like in the case of the newspaper.

The Semantic Web community proposed the use of ontologies and annotation as a necessary next-step in the evolution of web technologies, to allow the automation and enhancement of many tasks that nowadays are too costly to be performed in a manual way. Nevertheless, a manual creation of ontologies and the annotation of new and existing contents can also bear too high a cost. This problem places the need of semi-automation, as well, the creation of ontologies and the annotation of contents.

A rather different but interesting point of view that vindicates the need of automatic acquisition of ontologies is that given by Nirenburg and Raskin [2004]. Under discussion of the problem of *ontology granularity*[27], the authors remark that when manually defining ontologies a common aim is to avoid polysemy by splitting ontological concepts into the ever-smaller unambiguous unit. When using these finely grained ontologies to automatically annotate texts, the annotating tools will deal with the problem of disambiguating amongst too close senses. Therefore, it is desirable to acquire the ontology and annotate using the same text processing techniques so that the granularity of the ontology is adequate with respect to the capabilities of the tools to disambiguate and annotate.

2.3 Automated Ontology Acquisition

There is a wide variety of works in the literature that deal with the challenge of the automation of ontology acquisition and semantic annotation. Given that neither the use of ontologies nor the annotation of text against a knowledge model are exclusive for the area of the Semantic Web, but a resource used in other computer science fields (knowledge management, e-commerce, NLP, etc.), as introduced in Section 2.1, the approaches are diverse, sometimes following the specific goals of particular problems in each discipline.

Ontologies can be learnt from different sources.Gómez-Pérez et al. [2003] describe different methods and techniques to acquire ontologies from texts, dictionaries, knowledge bases, semi-

[27]Level of specificity of the concepts included in the ontology

structured data and relational schemata. Contreras [2004] proposes an architecture implemented by mixing different technologies (natural language processing for textual elements, layout processing to extract information from visual structures like tables or lists, data mining techniques) and data sources (web sites, textual documents or highly structured sources like XML files, tables, etc.). Most of the existing techniques have been focused on the processing of natural language texts or dictionaries definitions.

The systems for the processing of text may be classified in two categories:

- Building ontologies without an initial ontology

  In general, those methods that do not depart from an initial ontology or an initial hierarchy of concepts present the disadvantage of requiring a substantial amount of manual intervention to get good results. Sometimes the goal is limited to get a taxonomy, with no regards to other relations apart of the hyponymy [Bachimont et al., 2002]. In general, the approaches to build up an ontology without an initial seed ontology start the building process from a term glossary, defined by an expert in a manual way, or extracted from texts. The task of extracting terms from texts can be aided by some tool for the extraction of terminology based on natural language processing, whose results are afterwards revised by the expert. There is an important manual involvement in the overall process: defining relationships between concepts, finding and assigning explicit similarities and dissimilarities amongst neighbour concepts, giving natural language descriptions of the terms, etc. Some of the works that do not use a seed ontology are those of Aussenac-Gilles et al. [2000], Bachimont et al. [2002], Nobécourt [2000]. Some methods tried to introduce a higher degree of automation: the methodology proposed by Hwang [1999] requires a lower intervention than the methods mentioned above. It departs from a term glossary but the extraction of concepts and relations is automated, although limited. The ontologist has to supervise the final results from the data extraction process, but the author points out that this methodology does not deal in a proper way with the problem of language ambiguities.

  Some approaches in this category focus on learning some specific characteristics of an ontology, such as relationships between concepts [Davidov and Rappoport, 2008b,a], instance-of relationships [Van Durme and Pasca, 2008, Talukdar et al., 2008, Kozareva et al., 2008] or class attributes[Tokunaga et al., 2005, Pasca and Van Durme, 2007, Paşca, 2007].

- Building ontologies using a seed ontology or taxonomy

  Those methods that use a given ontology or taxonomy as knowledge source rely on automated techniques to learn new concepts and relations, and the human work involved is usually limited to the supervision of the results.

  Some of the approaches prune existing, general ontologies to be applied to specific domains [Missikoff et al., 2002, Lonsdale et al., 2002, Kietz et al., 2000, Gupta et al., 2002]. The most commonly used general-purpose ontology is WordNet [Miller, 1995]. These methods use general free texts and domain-specific texts to, by means of an statistical analysis, keep in the ontology only those terms that are relevant to the specific domain and discard

irrelevant terminology from the initial ontology, enriching it with new domain terms and relations if necessary.

In contrast to the previous approach, [Khan and Luo, 2002] proposes a clustering method whose aim is to obtain a hierarchy of concepts. To do so, some texts from the same domain are processed to extract a topic, and by means of clustering the topic terms are put in the correct place of a hierarchy. An external ontology (WordNet) is used to assign a concept to each node, in such a way that the assigned concept represents its corresponding clustered topic. Finally, some concepts from the general ontology are extracted to complete the clustered hierarchy. The main feature of this approach is that the domain hierarchy is created using clustering, rather than re-using an existing general hierarchy to prune it or to extend it to the domain. The general ontology is only used to complete the clustered structure.

Faatz and Steinmetz [2002] use a domain-specific medical ontology as initial data and enrich it using techniques similar to those used to enrich general ontologies. The main difference with the enriching approaches mentioned above is that they already depart from a domain-specific ontology.

Other methods enrich, extend or populate an initial general ontology with additional knowledge [Agirre et al., 2000, Alfonseca and Manandhar, 2002c, Gupta et al., 2002, Hahn and Schnattinger, 1998, Hearst, 1998, Kietz et al., 2000, Lonsdale et al., 2002, Moldovan and Girju, 2002, Pekar and Staab, 2003, Popov et al., 2003, Roux et al., 2000, Wagner, 2000, Xu et al., 2002, Snow et al., 2006, Pennacchiotti and Pantel, 2006, Suchanek et al., 2007]. In most cases, these methods process texts, that are in most of the cases domain-specific, to extract new concepts, relationships or other resources to enrich the initial ontology. The main feature of these systems is that the goal is the enrichment of existing ontologies rather than building them up, so the works are much more focused on the development of techniques like concept acquisition and classification, or relationships learning, and sometimes do not specify how the acquired knowledge is integrated in the given initial hierarchy. The work of Popov et al. [2003] is one of the most focused to the Semantic Web context. It uses the GATE[28] natural language processing tools, which have an very good performance in the extraction of entities from text. Suchanek et al. [2007] integrate WordNet and Wikipedia so they can be used jointly.

These methodologies gave way to the development of corresponding tools for knowledge acquisition and ontology build up or enrichment. Only a few of them (e.g. DOE[29], TERMINAE[30], TEXT-TO-ONTO[31], KIM[32]) export their results to Semantic Web languages like RDF, DAML+OIL or OWL. This is due to the fact that, thought applicable, some of these works are

[28] http://gate.ac.uk
[29] http://opales.ina.fr/public/
[30] http://www.lipn.univ-paris13.fr/∼szulman/TERMINAE.html
[31] http://ontoserver.aifb.uni-karlsruhe.de/texttoonto/
[32] http://www.ontotext.com/kim/index.html

not developed under the Semantic Web environment but arose from the natural language community[33]. All the tools require the supervision of an ontologist or an expert to guarantee an acceptable accuracy in their results.

2.4 Automated Semantic Annotation

Contreras et al. [2003] propose the following classification of systems for automatic semantic annotation:

- Ontology-based annotation tools.

  These tools identify and mark up, inside documents, instances of the concepts existing in an ontology. Therefore, they both populate the ontology with instances of the concepts, and annotate the documents. Some tools focus more on the population aspect, and others on the annotation. Both aspects produce the same, or similar tools.

  Some tools that address this task are MnM[34], AeroDAML[35], Onto-H and C-PANKOW[36].

- Wrappers.

  Wrappers use Information Retrieval, Information Extraction, natural language processing and machine learning techniques to extract information from different web sources under request, transforming data structured for human understanding to data processable by a machine. The input is some type of content source and the output is the information looked for. This information can be used as an input of a subsequent application (e.g. an agent) to exploit it (e.g. price information extracted from an on-line shop) or semantically annotated via meta-data. Wrappers do not necessarily use an ontology as knowledge representation model. Some examples of very well known wrappers are Google News[37], which aggregates news from over 4000 web sources, or CiteSeer[38], which is an autonomous citation indexing system which indexes academic literature in electronic format. Most of them perform a mapping from the original format of each source to a common, annotated format. In this case, the wrapper mostly exploits the *structure* of the source documents, taking advantage from repetitive schemes or text structures that help the semantic interpretation of the data. Only a few wrappers are mainly focused to the use of natural language processing techniques (e.g. Amilcare[39]).

[33] Specially with the works related to the enhancement of the lexico-semantic network WordNet
[34] http://kmi.open.ac.uk/projects/akt/about.html
[35] http://ubot.lockheedmartin.com/ubot/hotdaml/aerodaml.html
[36] [Cimiano et al., 2005]
[37] http://news.google.com
[38] http://citeseer.ist.psu.edu/
[39] http://nlp.shef.ac.uk/amilcare/

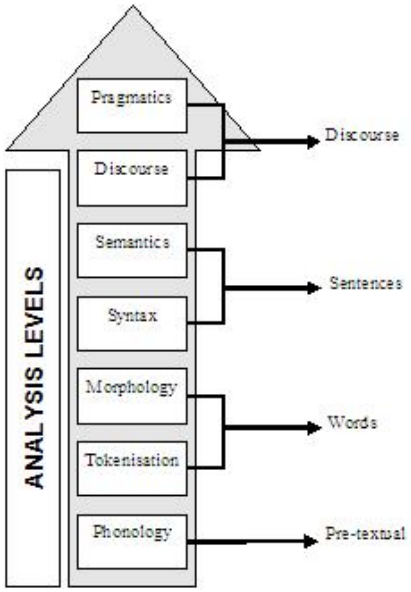

Figure 2.5: Levels in a complete Natural Language Processing application.

2.5 Analysis levels in Natural Language Processing

Natural Language Processing (NLP) is a field whose aim is the analysis and processing of human languages with computational techniques. Some of the practical applications of NLP are the implementation of dialogue interfaces, automatic Question Answering systems, automatic translation, natural language generation, Information Retrieval, Information Extraction, automatic summary generation and Text Mining, amongst other. NLP gathers different techniques, sometimes requiring the contribution from quite dissimilar disciplines: signal processing, to process a speech sequence and transform it to written text; linguistics, to characterise words, sentences and texts; statistics, to elicit the distribution of words in texts and understand their usage; even, for some levels of text analysis and some particular applications, NLP resorts to psychological patterns and common-sense rules.

Figure 2.5 represents a classical NLP system, which is usually decomposed into the levels of analysis described below [Engels and Bremdal, 2000, Alfonseca and Rodríguez, 2003, Contreras et al., 2003, Mitkov, 2003]. Each one of the analysis levels uses to be performed above the lower analysis levels, taking benefit from the previous steps.

Phonological analysis. The *phonological level of analysis* deals with the processing of the speech signal by means of speech recognition techniques to transform it into a sequence of textual words [Lemmetty, 1999]. Some applications, e.g., speech-based dialogue systems or spoken information systems, are intended to process a speech sequence stored as a sound signal with the goal of transforming it into text, which can be afterwards further processed using only the textual data or combining text and sound features. Some of the techniques used to this aim are Hidden

Markov Models (HMM), neural networks or mixed techniques [Rabiner and Huang, 1986, Colas, 2003].

Text segmentation. It basically consists of two main tasks: splitting the text in sentences and segmenting or *tokenising* the text into words.

Sentence splitting discovers whether a dot marks an abbreviation, a sentence end or both [Mikheev, 2002].

Concerning tokenisation, for those applications where the input to the NLP system was a speech sequence, in general, the speech recogniser will output a sequence of words, and therefore the tokenisation may not be necessary. When needed, the segmentation of text into words can be done using a set of tokenisation rules, usually written as regular expressions which describe how to match different types of tokens such as regular words (i.e. finding blanks in the text), numbers, punctuation, multi-word expressions, special expressions as Email addresses, date, time, etc. More advanced tokenisation systems provide facilities for composing complex regular expressions.

In the case of latin-, greek- or cyrillic-based writing systems, these methods work quite well, as word tokens are normally delimited by a blankspace. In the case of non-segmented languages, such as many oriental languages, identifying token boundaries requires more sophisticated algorithms like HMM or other statistical methods.

Morphology. The *morphological level* of analysis [Trost, 2003] consists in the processing of words to obtain its inner structure (lemma and affixes). A *stemmer* is a tool that performs morphological analysis to elicit the stem of a word from its modifiers, thus reducing as many related words and word forms as possible to a common *canonical form*. Usually a stemmer places a tag that identifies the inflected form: number, gender, case, verb conjugation, etc. Some stemmers analyse derivation and compounding of words as well, a more difficult task than processing inflections, as both derivation and compounding are processes that create new words, usually with a different grammatical category that the original ones.

Concerning the techniques used for morphology, some approaches use *declarative domain lexicons*, complete databases including all the possible inflected forms of a certain domain vocabulary, or the lemmas and the inflecting rules. The drawback of this approach is that the quantity of data to be stored depends on the number of words to be considered inside the domain vocabulary. Hence, it is limited either to languages that are not very highly inflected (e.g. English, Spanish, etc.) or to small controlled vocabularies.

For medium-large or general pursued vocabularies, some approaches are:

- Finite-state morphology, where the structure of words is seen as a concatenation of morphs that can be straightforwardly described by a finite automaton. This approach, nevertheless, cannot describe all the morphological phenomena.

- Two-level morphology, where rules are used to describe which combination of characters are legal words of a given language, filtering those morph combinations that are impossible

for that language.

- Continuation lexicons, where a lexicon classifies legal word formations by affixation. Given a sequence of text or speech, the lexicon is traversed from an initial morph to its legal continuation classes, until a stem word and its affixes are detected.

Either before or during the morphological analysis, the syntactic categories of the words (noun, adjective, verb, pronoun, adverb, preposition, conjunction, etc) are labelled. This task is known as Part-of-Speech (PoS) tagging. To assign the word categories, it is typical to use probabilistic models (n-grams, HMM) [Brants, 2000], Maximum Entropy models [Ratnaparkhi, 1998], and transformation lists [Brill, 1995], amongst other reported techniques.

Syntax. *Syntactic analysis* or *parsing* consists in examining sentences to analyse the grammatical arrangement of words inside them. It usually consists in identifying *constituents* or *phrases*, groups of words that share grammatical properties (they can be exchanged, and they can be used in similar places inside sentences), and the syntactic dependencies between them (e.g. predicate-argument relationships) [Kaplan, 2003]. These relationships can be used to represent the structure of a sentence as a tree, called *parse tree*, where the root represents the whole sentence.

The parsing of a sentence can be performed completely, by generating the complete parse trees (*deep parsing*), or it can be attempted in a more shallow (a simple) way. [Abney, 1991] was the first one to propose that parsing could be divided in several steps, each of which would focus on a particular kind of constituent. Later, a *shallow parser* may also recognise certain selected syntactic dependencies, e.g., between a verb and its direct object, or between a verb and its subject.

One of the most common shallow parsers are Noun Phrase chunkers. A *noun phrase* (NP) is a noun grouped with its modifiers (adjectives, determiners, etc.) A *NP chunker* is a tool that chunks all the non-overlapping non-recursive Noun Phrases found in text. NP chunkers can also be implemented in many ways, most typically by applying machine learning techniques trained on annotated corpora (e.g. maximum entropy models, HMMs, Support Vector Machines, transformation lists, memory-based learning, etc.) Some systems also use hand-coded grammar rules.

Some applications of NLP need a deeper syntactic analysis than NP determination or shallow parsing. This type of deeper analysis usually requires *grammars*, formalisms that model by means of rules the whole structure of a language, which can be defined under different approaches: regular grammars, context-free grammars, transformational grammars, constraint-based grammars, Head-Driven Phrase Structure grammars, etc. The grammars are used by a *syntactic parser* to produce the full syntactic analysis of a sentence, which can be implemented using different algorithms [Alfonseca and Rodríguez, 2003]: Top-down (departs from the sentence as a whole and chunks it while seeking syntactic dependencies, until getting single, analysed words); Bottom-up (reverse performance); chart parsing (uses dynamic programming to execute in polynomial time), etc. Very typical in practical applications is the use of statistical parsers Charniak [2000], Collins [2003], Nivre and Scholz [2004] trained on annotated corpora like the Penn Treebank Marcus et al. [1993].

Semantics. *Semantics* is the study of meaning. As in the case of syntax, the *semantic analysis* can be performed in a shallower or in a deeper way, depending on the final application.

A deep semantic analysis generally consists in translating a text written in human language into a non-ambiguous output format. This process is also called *semantic interpretation*. A classic way of performing this translation is based on the *compositionality principle*, which states that the meaning of a whole is a function of the meaning of the parts. In this way, we may assume that semantics is very related to syntax. One of the classic formalisms for representing meaning is the lambda notation, under which each syntactic constituent has a semantic value: e.g., proper nouns are constants, and verbs are functions that represent relations amongst nouns. "Meaning" is considered a logical rule pointing to a boolean truth value, $\{true, false\}$. As logical rule, the meaning can be interpreted to false or true by evaluating the functions and the constants in a sentence. Another technique to perform a semantic analysis is HPSG semantics, an approach where syntax and semantic analysis are fully integrated [Lappin, 2003]. A rather different approach for analysing semantics belongs to the machine learning field, which reports some inductive learning techniques to acquire semantic interpretations from annotated corpora [Ng and Zelle, 1997].

On the other hand, some applications require a simpler semantic analysis. In many cases the semantic enrichment required for a particular task consists only in making explicit the meaning of the words in a text. As introduced before, this is done relating the words with a corresponding concept in the knowledge representation model, by means of the semantic annotation. In general, the semantic analysis level required for annotation is that of solving problems arising from polysemy of words, the task of assigning to the ambiguous word the right sense in the ontology, which is called *Word Sense Disambiguation*. The different techniques to solve a disambiguation problem are deeply studied in Section 3.1.

Discourse. A discourse is an extended sequence of sentences that convey a package of information. The information within a discourse can include more than one topic. The analysis of the *discourse structure* of a text consists in fragmenting a text into discourse topics and studying how the different topics are organised in the text. *Discourse analysis* has the goal of marking topic boundaries, detecting the points in the text where a new topic is introduced, where an ongoing topic finished, re-introducing a previously treated subject, etc., and the relation between topics as per the discourse structure: enumeration, digression, contrasting topics, conclusion, etc.

The approaches to structure discourse include [Ramsay, 2003]: (a) Rhetorical Structure Theory, which states that a text can be decomposed successively in sub-parts, where each sequence of sentences are associated to one another by means of rhetorical relations, like *causal*, *purpose*, *motivation*, *background*, etc. The relations are recognised using heuristics, usually seeking cue words like "yet", "hence", "on the other hand", etc., and thus the discourse fragmentation can be analysed. (b) Centering theory, which says that each segment in a discourse features a topically most prominent concept, called "centre". It is possible to compute which term is the centre of a group of sentences and how this centre changes from one segment to another, quantifying if the central concept or topic continues, or is retained in a secondary position, from one group of sentences to another, or whether it sharply shifts.

Pragmatics. The analysis of a text up to its *pragmatic level* seeks to explain the meaning of linguistic messages in terms of their context of use. This is an important issue when implementing dialogue systems. It takes into account facts which are not explicit in the text, in the form of intended meanings of the speaker (*illocutions*) which can not be ascertained from a semantic analysis. It implies in most of the cases taking decisions about the participants in the communication act: the speaker's believes, his intention and the relevance of its assertions, as well as the listener's features. The pragmatic analysis can take advantage from the discourse analysis, as the rhetorical relations serve as indicator of the speaker's intention. There are different theories behind discourse pragmatics [Leech and Weisser, 2003]: (a) speaker centred theories, where the pragmatics of a discourse is determined from the effect that the speaker wants to achieve, (b) cooperative theories, where the pragmatics of a discourse are extracted from the co-operation (or lack of co-operation) of the speakers to the general goal of the conversation, (c) conceptual representations, where the pragmatics are determined from the effects in the hearer, amongst other.

Often *anaphora resolution* is considered a part of the pragmatic analysis. *Anaphora* is the linguistic phenomenon where a given *referent* is pointing back to a previously mentioned item in a text or *antecedent*. Anaphora resolution consists in determining the noun, noun group or clause being addressed by a pronoun or referent placed in a different sentence than its antecedent.

Depth of linguistic analyses in NLP applications. NLP can be understood as an end by itself, as a mean to interpret language and linguistic phenomena. Additionally, it is often used as a tool to assist other applications. The depth up to which the natural language has to be processed varies amongst the particular goals of the application:

Text segmentation and the morphological analysis level are quite useful for many applications and, probably, represent the tasks for which most automatic tools are available Brants [2000], Brill [1995]. Those applications that require deeper analysis levels usually start text processing with segmentation and PoS-tagging.

Deep parsing, as well as deep semantic interpretation, though widely applicable, is a costly task and thus has been applied in a lower extent. Apart from the theoretical linguistics interest, in general, deep semantic analysis is not so common in practical applications, with some examples being Answering (algorithms to automatically extract information to answer a given question) [Harabagiu and Moldovan, 2003], dialogue interfaces (systems capable to maintain an automatic dialogue with a user) [Androutsopoulos and Aretoulaki, 2003] and systems to assess free-text answers (evaluating free text user's answers, mainly for educational purposes) [Burstein and Leacock, 2000].

Word sense disambiguation is sometimes applied to those areas that need to cope with language ambiguities when interpreting text: Machine Translation (automatic translation of texts into other languages), Information Retrieval, Information Extraction, Question Answering, Text Summarisation (automatic generation of one or multiple texts summarising the original(s)), amongst other [Stevenson and Wilks, 2003]. These areas, as the Word Sense Disambiguation itself, can benefit also from shallow parsing.

Dialogue systems make use also of the discourse level analysis. Many contributions to dis-

course analysis come from the natural language generation community. Language generation systems deal with the problem of automatically generating grammatically well-formed, coherent, user adapted text from a knowledge source, for instance for information or educational purposes. These systems usually include a discourse planner and a grammar in their architecture [Milosavljevic et al., 1998].

Pragmatics is a challenging field with special applicability to the now emerging dialogue systems. Anaphora resolution techniques are also used for many applications, such as Information Extraction, Question Answering and free-text Computer Assisted Assessment.

Regarding the tasks in focus for this work, semantic annotation of text and ontology population, the linguistic analysis usually includes PoS tagging and word stemming, and sometimes NP chunking and other parsing. These linguistic analyses prepare the text for a later extraction of terms, resolution of ambiguities, determination of semantic relations and final annotation or classification of concepts in the ontology.

Chapter 3

Semantic Enrichment with Natural Language Processing

This chapter reviews the state-of-the-art in Natural Language Processing techniques that are relevant to the research described in this thesis: Word Sense Disambiguation (WSD), Information Extraction (IE) and Relationships Extraction (RE). A final section focuses on the applications of these techniques to annotating semantic information in the Wikipedia.

3.1 Word Sense Disambiguation

3.1.1 Introduction

Ide and Véronis [1998], define Word Sense Disambiguation as the task of associating a given word in a text or a discourse with a definition or meaning (sense) which is distinguishable from other meanings potentially attributable to that word.

As stated in Chapter 1, one of the challenges of semi-automatically annotating a web page is how to process web content written in natural language, specifically, how to solve the ambiguities of polysemous words found in a web page, when annotating it. The WSD task, in the SW context, is redefined as the association of the polysemous word to one of the senses under which this word appears in a background ontology.

To discriminate among the possible senses, a WSD technique may rely on two major sources of information:

- The context of the polysemous word
- An external knowledge source

A brief explanation of how these two sources of information are used for disambiguation is the following: once a polysemous word is found in the text, a certain context surrounding the word in the text is chosen, and the words in this context are contrasted with the information contained

in the external knowledge source in the search of evidences to discriminate among the possible senses.

3.1.2 The Use of Context in Disambiguation

The *context* of the word to be disambiguated plays a important role when attempting to align the word with one of its possible senses. The context of a word consists in the words that co-occur with it, and is often limited to the length of the single sentence where the ambiguous word appears, or to a certain number of words regardless of whether they match a complete phrase, e.g. *n-grams*, *n* words to the left and/or the right side. Nevertheless, the context can also include other words within the same text or discourse, as well as extra-linguistic information about the text such as pragmatic data (situation, etc.), or the text topic.

3.1.2.1 Classification of contexts used in Word Sense Disambiguation

When processing the context, the amount of information that can be kept from the context varies:

- Bag-of-words approach: The words in the contexts are represented as a group, without consideration to the syntactic relation between the words or to the ordering of the words inside the sentence. This is the main approach used when the context considered is large, containing more than one sentence. Under this approach, [Miller et al., 1994] defines *co-occurrence* as the occurrence of two words inside a context of a given length. The methods using the bag of words approach usually keep in the context bag only *content* words[1] , or *open-class* words, that is, they remove everything except nouns, verbs, adjectives and adverbs.

- Relational context approach: Additionally to keeping the words that co-occur with the polysemous word, some relations between the words are also recorded, such as syntactic relations, phrasal collocations, etc. These relations between context words are usually used in some WSD approaches that process sentence-long contexts. Yarowsky [1993] defines *collocation* as the co-occurrence of two words in a defined relationship:

  - Direct adjacency
  - First content word to the left or to the right of the polysemous word with same part of the speech.
  - Syntactic relation verb/object.
  - Syntactic relation subject/verb.
  - Syntactic relation adjective/noun.
  - etc.

[1]Content words or *open-class words* are nouns, verbs, adjectives and adverbs. *Close-class words* are the rest grammatical categories: determiners, prepositions, etc., which have a functional role in the sentence and usually do not convey semantic information

The works in [Yarowsky, 1993] show that certain syntactic relations seem to disambiguate better words with a certain Part-of-Speech, for instance: verbs derive more disambiguating information from objects than from subjects, adjectives derive almost all of their disambiguation information from the nouns they modify, nouns are best disambiguated by directly adjacent adjectives or nouns, verbs are less useful for noun disambiguation.

[Ide and Véronis, 1998] differentiate three types of context depending on its length relative to the whole text:

- Microcontext: Also called "local context". The context considered is a small window of words surrounding the ambiguous word. It usually ranges between 1-2 words offsets to 20-50 words at each side of the polysemous word. [Yarowsky, 1992] uses a window of 100 words and points out that the required size depends upon the function of the polysemous word in the sentence, finding that prenominal modifiers (adjectives and compound nominals) are heavily dependent on the noun modified, usually inside the same sentence, so a much narrower window would be enough. [Vorhees, 1993] uses a disambiguation approach based on hyponymy relations (see in Section 3.1.4 the methods based on a network structure) for an Information Retrieval application, finding that the extremely short context in a retrieval query doesn't work well with methods based only in this type of relations, and that a longer context should be required. [Sussna, 1993] uses a window of 30-40 words and points out that the information given by the context is useful when considering a single topic, and therefore the context length would depend upon the discourse fragmentation. This consideration leads us to the next type of context:

- Topical context: Also known as "global context". The topical context includes several sentences within a same discourse. The size of a discourse varies depending on the type of text under consideration, e.g., a text extracted from a newspaper uses to be structured in shorter discourses than, say, the full-book discourse of an academic, technical text of a very specific field. The contribution of the segmentation into topics to the task of disambiguation is, nevertheless, not well studied, and there are results that indicate that, for some approaches, microcontext alone is superior to topical context as an indicator of sense.

- Domain: Some approaches take into consideration the general domain under which the text could be classified, and use it to discriminate senses. For example, in a text about Royalty, the word *queen* will probably be used referring to a female head of a Kingdom.

  Dahlgren [2000], however, remarks that some highly polysemous words (e.g. *hand*) retain most of their senses in almost any domain. In this cases, some wrong assignments can be made. Following the above example, if there is any reference in the text about Royalty making reference to the chess piece *queen*, it will be mis-tagged because the main domain of the text is different. [Yarowsky, 1992] points out that very polysemous words use also to be very frequent, and part of that frequency is due to the use of idioms. For instance, Ide and Véronis [1998] mention 16 distinguishable senses of the word "hand", 10 of them can be used in almost every type of text. Interestingly, a good fraction of its usage comes from

idioms (e.g. "on the other hand"), which have only a functional role in the sentence. Many lists of idioms exist (e.g. in Roget's thesaurus) and these can be easily recognised.

3.1.3 A Classification of External Knowledge Sources for WSD

The *external knowledge source (EKS)* is a repository of information that helps the disambiguation task. It usually includes lexical, semantic and/or syntactic information about how the language is commonly used and what words constitute the vocabulary to take into consideration. Somehow, like the ontologies for the Semantic Web, these EKS constitute a conceptualisation of the knowledge and some models rely on general theories about language [Chomsky, 1957]. The structure taken by the EKS used for disambiguation varies a lot depending on the WSD approach: whilst some of them are built *ad hoc* for the disambiguation task (see below the *word expert* approaches), others are very general models that, through a convenient processing, serve as a basis for many tasks apart from disambiguation, e.g. machine-readable dictionaries. There are models that share a high similitude with the ontologies used for the Semantic Web, so they are very useful for the aim of this work as they can be easily ported to a Semantic Web format (see below the structure of the WordNet lexico-semantic network). Regarding the lexical coverage, there is again a wide range: from knowledge sources for only a small portion of the language, limited to specific domains or even to a few words, to EKS covering general language terminology.

Regarding the particular techniques used to carry out a disambiguation task, the early works in Machine Translation, dated from the fifties, already outlined most of the keys being in use nowadays. At that time, the quantity of data that could be managed by a computer was limited, and the evolution of the hardware in the last decades of 20th century, specifically the increase in the storage capacity, gradually gave way to more and more complete EKS. Like in the case of the ontologies for the Semantic Web, nowadays the main limit to the size of an EKS depends on the amount of manual work that its construction requires rather than on hardware capabilities.

Ide and Véronis [1998] make a classification of WSD methodologies mainly based on the form that the External Knowledge Sources took along time. Three main groups can be differentiated:

3.1.3.1 Artificial Intelligence lexico-semantic networks

These systems began to flourish in the sixties and were the object of extensive study during the next decade. Some of the systems were in use and under continuous development up to the late eighties. The knowledge source was typically grounded in some theory of human language understanding, and often involved a great effort to manually define extremely detailed knowledge about syntax and semantics.

The knowledge underlying the data is modelled by means of a lexico-semantic network, where the meanings of the words (concepts) are represented as the nodes of a graph. Similar senses are closely related in the network, and, generally, there is a hierarchy between concepts in a way that, starting from one single node, the rest of the concepts are arranged as a tree of superconcepts-subconcepts, while superconcepts subsume the sense of their subconcepts. The terms in the hi-

erarchies are furthermore linked by semantic relations, or by some kind of association between words, e.g. verb senses, represented as arcs in the graph. The methods that use lexico-semantic networks exploit the hierarchical relation and the usually rich range of word associations to discriminate among senses in a context. The main difference between these lexico-semantic networks and the computational lexicons developed in the mid-eighties (introduced below) is that these early networks were usually made *ad hoc* for each specific disambiguation approach, and there is a high amount of human intervention required for such a specific task. This problem reduced the WSD experiments to the scope of small contexts containing those few words for which the network was highly enriched with features.

3.1.3.2 Knowledge-bases

The availability, in the eighties, of large scale resources as machine-readable dictionaries (MRDs), thesauri and corpora, allowed the automatic extraction of knowledge from them.

- Machine Readable Dictionaries. The electronic version of general purpose dictionaries opened a way to overcome the problem of manually defining the complex semantic networks required by the AI approaches. Some of the first electronic dictionaries used for WSD were the *Collins English Dictionary* (CED) and the *Longman Dictionary of Contemporary English* (LDOCE). The text of the definitions was used in a very direct way, as will be seen later, disambiguating by merely contrasting definition words with context words [Lesk, 1986]. Some approaches tried to make an intermediate step, transforming the dictionary knowledge into a lexico-semantic network format, but this goal was not fully achieved due to the inconsistencies found in dictionaries: there is a handicap in the fact that the words inside a dictionary definition are ambiguous themselves, and the dictionaries were originally written for human use, in such a way that although the senses are well discriminated as dictionary entries, these senses are used indistinctly inside the definitions.

  Wikipedia deserves a special mention as a dictionary/encyclopedia-based resource. Mihalcea and Csomai [2007] use it both as a repository of senses, and as an annotated corpora, by automatically identifying important terms in the entries and introducing hyperlinks between these terms and existing entries. Similar approaches are those by Medelyan et al. [2008b] and Milne and Witten [2008].

- Thesauri. Thesauri are hierarchies of categories, where a set of words in a same category are semantically related. The most common relationship amongst words included in a thesaurus is synonymy, although other relations can also be found. The relation that defines the hierarchy is the hyponymy between categories of words. Some of the thesauri used for disambiguation were *Roget's International Thesaurus*, the *Hallig-Wartburg* in German, or the *Longman Lexicon of Contemporary English*. The hierarchy of categories in a thesaurus can be used to discriminate among senses, using methods that take into account the hierarchical IS-A relation, as will be seen later. Nevertheless, as in the case of dictionaries,

thesauri were created for humans and they do not reflect the relations between words in a perfect, error-free way.

- Computational Lexicons. Like the semantic networks used in the AI approach, computational lexicons are knowledge bases constructed by hand. These resources are large-scale knowledge bases, sometimes built by the collaborative effort of different institutions, for general NLP purposes, unlike the "toy" implementations handling a tiny fraction of the language, used for very specific applications, under the AI approach. Computational lexicons usually include many of the common resources exploited in the disambiguation field, combining dictionary definitions, groups of words arranged in a hierarchy like a thesaurus and semantic relations, at least. The lexicons are built either in an *enumerative* way, where senses are explicitly provided, or in a *generative* way, in which the senses are generated using rules. The most widely used lexicon is WordNet [Miller, 1995]. WordNet was conceived as a lexical semantic network. It structures the entries according to word meaning in a semantic network where each node contains a synonym set or *synset* that represents a certain *concept*. All the words in a synset express the same meaning. For each synset, a *gloss* gives a brief definition of the concept in natural language. Several lexico-semantic relations are included in the network. The dominant relation is hyponymy/hypernymy, which structures the concepts into hierarchies. In the first versions of WordNet, only a small set of additional relations was added: meronymy (HAS-PART) and its inverse holonymy (PART-OF or MEMBER-OF relation), antonymy for nouns, as well as several relations between verbs and adjectives. Later on, revised versions and extensions of WordNet have included more relations, so for example version 2.1 contains the following ones:

  - For **nouns**: Antonym, Hypernym, Instance Hypernym, Hyponym, Instance Hyponym, Member holonym, Substance holonym, Part holonym, Member meronym, Substance meronym, Part meronym, Attribute, Derivationally related form , Domain of synset (topic, region, usage), and Member of this domain (topic, region, usage).
  - For **verbs**: Antonym, Hypernym, Hyponym, Entailment, Cause, Also see, Verb Group, Derivationally related form, and Domain of synset (topic, region, usage).
  - For **adjectives**: Antonym, Similar to, Participle of verb, Pertainym (pertains to noun), Attribute, Also see and Domain of synset (topic, region, usage).
  - For **adverbs**: Antonym, Derived from adjective, and Domain of synset (topic, region, usage).

  WordNet was also extended with additional knowledge resources. Some of them are domain labels for each synset [Magnini and Cavaglià, 2000, Bentivogli et al., 2004], a distinction between concepts and instances following the Semantic Web model for ontologies [Alfonseca and Manandhar, 2002a], or topic signatures [Agirre et al., 2001]. The disambiguated gloss relationships are also available for WordNet 3.0[2].

[2]http://wordnet.princeton.edu/glosstag

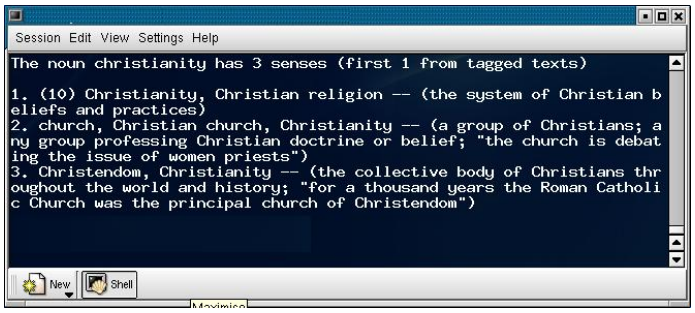

Figure 3.1: Example of granularity for sense distinctions in WordNet

There are versions of WordNet in many other languages, such as Dutch, Italian and Spanish [Vossen, 1998], German (GermaNet), French, Czech, Estonian, several Balkan languages [Tufis et al., 2004], Oriya [Mohanty et al., 2002], Tamil [Poongulhali et al., 2002] and Hindi [Chakrabarti et al., 2002], to cite some of them. Additionally, it has been ported to RDF[3].

Recently there have been several efforts to extend WordNet with encyclopedia information, which can possibly be mined, like the work from Ruiz-Casado et al. [2005a] (part of this thesis), or Suchanek et al. [2007]

Although being the most widely used computational lexicon, WordNet has also its drawbacks: it has been criticised because of the fine granularity of its sense division. Some of the senses are so finely distinguished that the differences are difficult to be seen, even by a human. Fig. 3.1 shows the fine distinguished definitions of *Christianity* in WordNet 1.7. This fine granularity makes the WSD a hard task, as very close senses use to be subsumed by close common ancestors in the hierarchy, and also co-occur with the same words in different contexts, so the disambiguation keys fail to differentiate them. A solution could be to group WordNet senses, but the extent up to which this grouping should be done is again a source of disagreement, and in general it will be task-dependent: for disambiguating general texts, like news, the grouping should be more severe than for domain specific tasks, where the WordNet sense distinctions can even be too broad (think in the above *Christianity* example used in a text from the field of theology). Furthermore, some of the relations are not pivoting over the same feature: for instance, *man* is found to be an antonym of *woman*, as *software* is found as antonym of *hardware*, which is a subjective antonymy definition, in contrast to the most commonly agreed antonymy relationship between adjectives or adverbs, like *far / near*, *soon / late*, *long / short*, etc.

Some other hand-crafted computational lexicons are Cyc [Lenat and Guha, 1990, Lenat, 1995], ACQUILEX [Copestake, 1992], COMLEX [Macleod and Grishman, 1994], and CORELEX [Buitelaar, 1998]. Cyc, in particular, includes rules that allow the inference of relations between words, and tools for analysing time and temporal classification of events in a text. Cyc was too richly specified at their birth, and the human efforts to match these

[3]http://www.semanticweb.org/library/

severe specifications were too high. They didn't have the same diffusion and didn't take profit from massive collaborative enrichment as WordNet did, and therefore have not been used to the same extent.

3.1.3.3 Annotated Corpora

A corpus is a collection of documents written in natural language where the words are tagged with their corresponding senses. In this way, a corpus is a bank of *samples* that enables the extraction of sense-discriminating rules and disambiguation metrics based on the statistics of the senses assigned to the words in the corpus. One of the main problems of methods based on annotated corpus, that is to say, methods based on encountering an annotated example to apply it later on, is the so-called *data sparseness problem*. Natural language is so rich that a same thing can be expressed under many textual forms: a specific sense of a very polysemous word can be rare in a corpus extracted from, say, an encyclopedia, or a news repository, and even in the case that it is found, the way it is used can be only one of the many possible ways to use that word with that sense. To ensure that all the senses and all the possible ways to use a word are represented in a corpus, the amount of documents required to be manually annotated have to be very high. The problem of those NLP approaches that base their performance in having an example of the use of the word is that they can not process words that are not found in the corpus, for which tagged data is not available. Corpora are normally built by hand, and this constitutes a handicap. Several efforts have been made to automatically annotate corpus via *bootstrapping*, a technique based on using some few examples as a seed to learn rules for tagging. To overcome data sparseness, some methods combine thesauri with corpus and learn rules that apply to words in the same category, in such a way that when a new word appears, even if there is no example of usage for this word, the rules applying to its category are considered to be applicable to the new word [Brown et al., 1992]. Nevertheless, the hypothesis that all the words in the same class behave in a similar fashion has been found too strong, and fails to be true in many cases.

Not really an annotated corpora itself, Wikipedia has also been used as a source corpus for training a Word Sense Disambiguation system [Mihalcea, 2007]. By collecting sentences containing link anchors to entries that are mapped to WordNet synsets, it is possible to collect a set of sentences where at least one of the terms is disambiguated using WordNet as a repository of senses.

3.1.4 A Classification of Features and Techniques for WSD

The sections above have introduced some classifications for contexts and EKS. As far as the diverse WSD approaches use a combination of a particular type of context and external knowledge source to select the right sense of ambiguous words, each particular technique will, of course, have a high dependency upon the specific context and EKS used for the application. The range of combinations is broad. Some authors classify the approaches as per the EKS [Ide and Véronis, 1998], or divide the approaches per their ability to infer new knowledge. [Mihalcea et al., 2004b] classifies two types of methods: those, mainly based on annotated corpus or other type of hand

tagged data, that are applicable only to words that have been previously found in the corpus, and those methods that can make logical inferences or extract global properties and therefore can extend the data available in the EKS with new knowledge. For instance, some methods use the relations in a graph-based EKS (e.g. a computational lexicon) in such a way that the words representing a certain context, or a certain sense, can be extended with their hyperonyms or other words connected to them by relations to try to help the disambiguation.

Given that most systems for WSD nowadays are based on machine-learning techniques, trained on annotated corpora, most of them use a variety of features obtained from different resources. This section describes these features grouped according to which is the source from which they can be obtained. Some of them were used in the past in a standalone way in complete systems, while others have only been considered in combination with others.

The following are the main data sources from which predictive features for disambiguation can be obtained:

- EKS network structure
- Linguistic information
- Frequency based statistics
  - In dictionaries
  - In thesauri
  - In annotated corpora
- Rules and patterns

3.1.4.1 Features obtained from the structure of lexico semantic networks

Some features for WSD can be obtained from network-shaped EKS, either the early lexico-semantic networks from the AI approach or modern computational networks, as described above. Their disambiguation performance usually exploits the network structure by using the links between concepts, commonly representing relations between concepts, to infer new knowledge. For instance, a system may look for the parents or upper ancestors in the hierarchy of a context word, in such a way that new knowledge that was not present initially in the context (these ancestor words) is being inferred thanks to the EKS structure and used for the disambiguation purpose. These methods will be also referred to as relations-based methods, as they follow the links in the EKS while processing the features encoded therein.

Some of the techniques using the network structure are the following:

- *Semantic similarity based on distance*:

  There is a wide variety of approaches that compute the semantic similarity between words by means of a "distance" measure. This distance is measured over a "path" of relations that link the possible senses of the context words in the hierarchy.

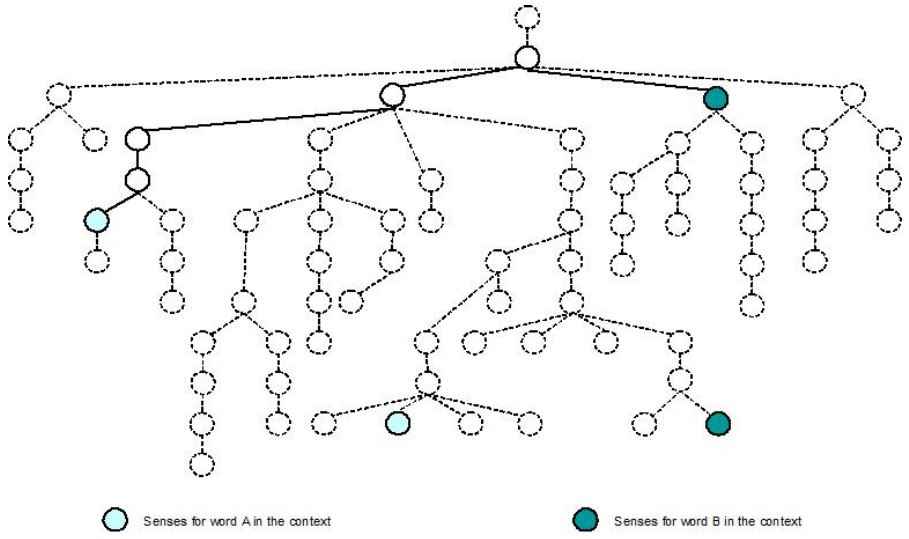

Figure 3.2: Minimal path for two binary sense words.

The first works using the positional distance of terms in a network used the *minimal path*. When two words co-occurring in a same context are presented to the network, the paths of arcs connecting all their possible senses in the network are pairwise calculated, and finally the senses are assigned to those nodes corresponding to the most direct path between two senses (minimal path) found in the ontology. Fig. 3.2 shows an example.

Quillian [1969] presents a disambiguation technique (Marker Passing) based on the approach presented above. In his work, the EKS is an *ad hoc* hierarchy of concepts linked by the IS-A relation. Additionally, many other relations are defined for each concept: predicates stated by a verb phrase or a relative clause, or any set of adjectival or adverbial modifiers. The knowledge source is furthermore complemented with manually defined syntactic patterns that model the usage of the word senses in common sentences. The context considered for disambiguation is limited to short sentences. WSD is accomplished navigating through the EKS, marking all the nodes whose relations depart from the possible senses of the context words, marking next the words pointed from these new nodes, and so forth. Whenever a previously marked word is found, an *intersection point* is defined, which is a point upon which a path in between two input words pivotes. The intersection point in a path is considered an ingredient common to the two initial words, and the minimum path connecting two senses is an indicator of relatedness, as very unrelated terms will intersect far away in the hierarchy. The candidate senses selected through the minimum path are later on checked against the syntactic patterns included in the data base, to confirm that the senses selected are right.

In a very similar way to the previous approach, in [Dahlgren, 2000] the network is traversed to find *common ancestors* for the words in the context, which determines the ontological similarity between senses. The common ancestors are equivalent to Quillian's intersections.

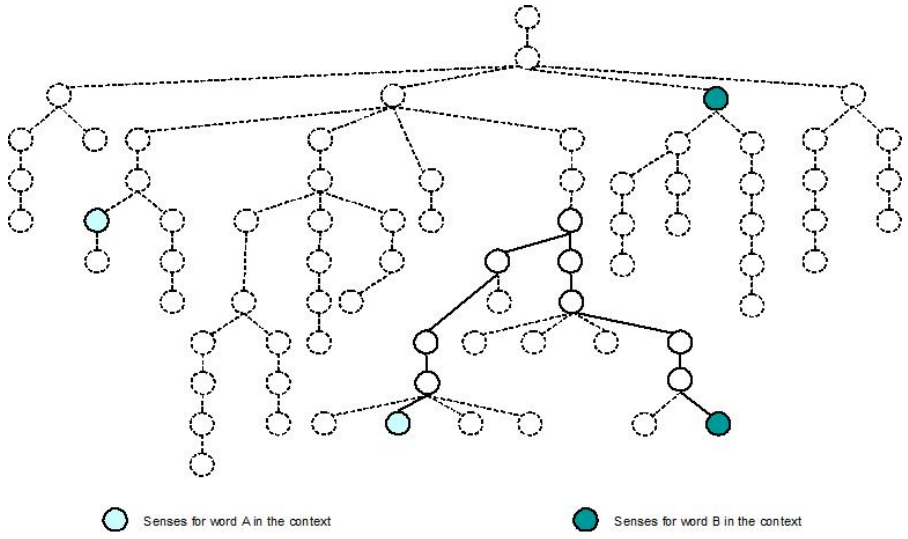

Figure 3.3: Most specific common ancestor for two binary sense words.

Later on, distance based methods were enhanced by assigning weights to the paths. One of the approaches, [Sussna, 1993], assigns a certain weight to each WordNet link depending on the type of the relation between the words (synonymy, hyponymy...), and a distance metric uses these weights to calculate a *semantic distance* between WordNet senses, computing it through the path that connects each pair of senses. The weights for each type of WordNet's relations are set in such a way that synonymy receives a distance of 0 (it relates representations of the same concept) and antonymy receives the maximum distance. The IS-A and PART-OF relations receive an intermediate weight ranged according to the number of arcs departing from each node. When calculating the distance between two nodes, the algorithm takes into account not only the weighted relations, but also the hierarchical depth of the terms. By doing so, when two terms being compared are found to be in the bottom of the hierarchy, their relatedness is reinforced, as it means that there is a relation between very specific terms rather than the expected relatedness between general terms. To disambiguate, given a set of words co-occurring in the context, the senses selected will be those that minimise the semantic distance between them.

Some other approaches mix the use of the IS-A relation in WordNet with empirical probability estimates. [Resnik, 1995a,b] claims that the semantic distance is not uniform for a same WordNet link type, e.g., *rabbit ears* IS-A *television antenna*, a hyponymy relation found in WordNet, intuitively covers a narrower semantic distance than *phytoplankton* IS-A *living-thing*, also found in WordNet under the same type of relation. Methods relying in the minimal path that connect two words and similar ideas, have the problem of considering a uniform distance for a same type of link. Resnik deals with this problem associating to each concept the probability of finding it, or its subconcepts, in a general text. These probabilities are extracted, in Resnik's experiments, from the Brown Corpus. As far as a

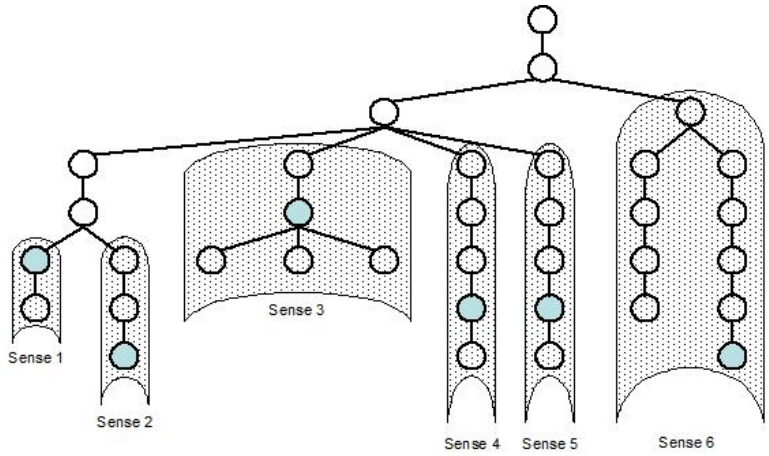

Figure 3.4: Part of a hierarchy with hoods for a 6-sense polysemous word.

superconcept always subsumes the meaning of its subconcepts, the extracted probabilities for each concept are increasing when following the taxonomy from the bottom to its root. By assigning the minus logarithm of the frequency, the values range from zero at the top of the hierarchy to a maximum value in the bottom. The minus logarithm comes to represent the "information content" of each concept: as probability increases, informativeness decreases, so the more abstract is a concept, the lower its information content. Then, the similarity between two concepts is computed by looking for the most specific common ancestor (see fig. 3.3), but taking into consideration the information content of each concept to make the distance not uniform for a same link. This way, two terms will be highly related semantically when the most immediate common ancestor is a very specific term, and two terms can be very away semantically when the common ancestor is too general, it can even be the root of the ontology.

- *Hoods*: [Vorhees, 1993] uses a combined technique for the disambiguation, which relies both in the hyponymy relation represented in the WordNet hierarchy and in the frequency of the words in the context. The author defines a "hood" for a certain sense S of a polysemous word W as the WordNet's subgraph that contains S, its ancestors and descendants, but does not contain any other sense of W apart of S (see fig. 3.4). The idea behind a hood is that it represents a grouping of words that embrace one of the senses of W, and every hood represents a different sense of the word. Following this idea, the sense of an ambiguous word in a given context can be selected by computing the frequency of the words in the context that occur in each of its possible sense's hoods.

- *Spreading activation models*: this approach mixes linguistic information, an approach presented below, with the exploitation of the concept network structure. It is based on the idea of the *semantic priming*, in which the introduction of a certain concept in a context will influence and facilitate subsequently introduced concepts that are semantically related. The

concepts in a semantic network are activated upon use, and the activation spreads to other nodes, e.g. the concept *throw* will activate the "physical object" sense of the word "ball". With the spreading, the activation is gradually weakened, but at the same time some nodes in the network can receive activation from different sources and result in the best sense for the ambiguous word. The concept of inhibition is also introduced, following the notion that the activation of certain nodes will also inhibit the activation of certain neighbours, e.g. in the former example, the activation of *throw* will inhibit the "social event" sense of the word "ball", while activating the "physical object" sense [McClelland and Rumelhart, 1981].

A similarly inspired work, but using Wikipedia as the repository of word senses, and its hyperlink structure as the network, is that described by Milne and Witten [2008], where the similarity between two word senses (represented by their Wikipedia entries) is calculated as a function of the distribution of in-links from other entries.

3.1.4.2 Linguistically-motivated features

These features are obtained from linguistic information stored in the EKS regarding how the words in a language can be combined.

- *Word Expert Parser*: it is based on the background theory that claims that the human knowledge about language is not organised as rules but as words, and therefore understanding should not be based on the matching of rules but on the interaction between words. This approach [Small and Rieger, 1982] requires the manual specification for each sense of a word in a *word expert discrimination net*, which includes the particular syntax and semantics that characterise the use of that sense against the rest of senses, thus allowing the *discrimination* among senses. When disambiguating a word, the selection of the corresponding sense is found by traversing the discrimination net with information from the word's context and information from other word experts. The context for disambiguation is usually a sentence, which is processed word by word, from left to right, activating at each step the discrimination net for each instantaneous word. As the different experts are activated, all the possible concepts for which the word in focus could be an instance are considered. The experts follow a decision logic based on throwing messages to the rest of activated experts and processing replies. There is an extensive exchange of questions and answers in the algorithm, usually elapsed in time until the activation of subsequent word experts in the sentence. The information exchanged is oriented to fill in some slots of "facets" that characterise each particular concept, like its constraints, or the role they have in the sentence (in an adjective, which noun it modifies, in a noun, if it is the object of a verb, etc.). All this interactive process is kept in order to refine the initial sense possibilities up to the determination of each single concept that is being represented by any of the words, and it does not stop until the whole initial sentence has been comprehended.

- *Selectional restrictions*: also called *preference semantics*, is a case-based approach where certain restrictions for the combination of items in a sentence are given. For instance,

whenever the verb *to drink* appears in a sentence, the selectional restrictions would indicate that an animated subject is preferred, and the object should be a hyponym of the name *drink*. The restrictions can be relaxed to handle phenomena like metaphor or metonymy, for instance, in the previous example the selectional restrictions can indicate that an animated subject is preferred but, if it is not found in the sentence, allow an inanimate one in order to handle expressions like *My car drinks gasoline*.

Wilks [1985] introduced this approach, using an external data base where the words (nouns, verbs) are defined by their lexical form and linked by a predicate to the type of arguments with which each particular word sense can be considered. The type of the arguments are qualitative (human, recipient, place...), and the type of the connector between words (direction connector, e.g. *to, from*, location, e.g. *in, at*, etc.) are also provided. The context used for disambiguation is formed by a sentence. The sentences are disambiguated by applying the predicates to the words they contain, to check which senses match the types of arguments defined in the predicate.

3.1.4.3 Frequence-based features

Some methods are based on the distributional properties of words, commonly representing context as a bag-of-words, that is, a group of words co-occurring with the ambiguous word, regardless their relative collocation in the context. The co-occurring words are searched in the EKS in order to extract some frequency-based metric, that is used to compute similarity between the target word (the word to disambiguate) and its possible senses.

For instance, the above mentioned approach [Resnik, 1995a] is a hybrid implementation of a relations-based method and a frequency-based method. Like in the Resnik's implementation above, or in the methods using thesauri, whenever the EKS is a taxonomy these methods are usually combined, so the frequencies are somehow mixed with the relations in the taxonomy to expand the frequency-based metric to other words in the hierarchy, hence making an inference of new data.

Disambiguation using the Vector Space Model When using frequency based statistics to process texts, a very common representation of the possible senses of the words and the context where they appear is the Vector Space Model (VSM), a term adopted in NLP from the Information Retrieval field. The disambiguation using the VSM is normally performed in two steps, (1) modelling the polysemous word and its sense candidates with vectors and (2) comparing the vectors to disambiguate.

The first step consists in creating one vector for the polysemous word and as many vectors as possible senses. In the model, each one of the vector's dimensions represents a word in the vocabulary, and each dimension receives initially a value equal to the frequency of that vocabulary word inside a context.

This way, the vector representing the polysemous word has positive values in the dimensions corresponding to the words found in its context, and the vector that represents each candidate

sense has positive values in the dimensions corresponding to the words related to that particular sense. For instance, the vector associated to a sense may be calculated from the content words in its dictionary definition, or from the words in a thesaurus category for that sense.

Most of the cases, systems do not work with raw frequencies. Rather, for each dimension in the vector, a weight is calculated, generally rating the importance of that word in the modelling of the concept. The statistics used to calculate weights are diverse. Some of them are:

- **tf·idf**: this metric combines the term frequency (tf) with the inverse document frequency (idf), that is, the inverse of the number of documents that contain that term. Therefore, if a term is at the same time very frequent in a context vector, and it does not appear in the other contexts, it will receive a high weight, indicating that it is a very relevant term.

$$weight_{ik} = tf_{ik} \cdot log_2 \frac{N}{n_k} \tag{3.1}$$

  where tf_{ik} is the frequency of the context word c_k in the context of sense s_i; N is the total number of senses; and n_k is the number of senses such that c_k appears in their context at least once.

- **The χ^2 function**: let us suppose that we have a set of senses $\{s_1, ..., s_N\}$, and a set of context words $\{c_1, ..., c_M\}$. Let's call $freq_{ik}$ the frequency of word c_k in the context of s_i. Then, the expected mean m_{ik} is defined as

$$m_{ik} = \frac{\sum_i freq_{ik} \cdot \sum_k freq_{ik}}{\sum_{i,k} freq_{ik}} \tag{3.2}$$

  The χ^2 weight for context word c_k in the context of s_i is then

$$weight_{ik} = \begin{cases} \frac{(freq_{ik} - m_{ik})^2}{m_{ik}}, & \text{if } freq_{ik} > m_{ik} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

- There are others, such as the $t - score$ [Church et al., 1991], based on the $t - Student$'s test, and the Mutual Information [Hindle, 1990].

Principal Component Analysis is a typical procedure for reducing the dimensionality of the data and automatically finding correlations between the different features. In the case of the vector space model, the use of Latent Semantic Analysis (LSA) is also very common and has been proved to improve the accuracy of the classification in many instances.

The final step is to compare the context vector with each of the candidate sense vectors to select the most similar, and to do so several similarity metrics between vectors are used. Some of them are [Matsumoto, 2003]:

- The dot product: given two vectors, x and y, of N dimensions, it is defined as

$$sim(x, y) = \sum_{k=0}^{N} x_k \cdot y_k \tag{3.4}$$

- The cosine of the angle between the two vectors is one of the most commonly used metrics:

$$\begin{aligned} cosine(x, y) &= \frac{x \cdot y}{|x| \times |y|} \\ &= \frac{\sum_{k=1}^{N} x_k \times y_k}{\sqrt{\sum_{k=1}^{N} x_k^2} \times \sqrt{\sum_{k=1}^{N} y_k^2}} \end{aligned} \tag{3.5}$$

  In practise, it is a normalised dot product.

- The Kullback-Leibler divergence:

$$KL(x, y) = \sum_{k=1}^{N} x_k \log \frac{x_k}{y_k} \tag{3.6}$$

- The Jensen-Shannon divergence:

$$JS(x, y) = \frac{1}{2}[KL(x, \frac{x+y}{2}) + KL(y, \frac{x+y}{2})] \tag{3.7}$$

- Jaccard's coefficient:

$$Jaccard(x, y) = \frac{|\{i : x_i > 0 \wedge y_i > 0\}|}{|\{i : x_i > 0 \vee y_i > 0\}|} \tag{3.8}$$

We will next present some approaches that use frequency based statistics to disambiguate, classifying them as per the EKS from which the candidate senses for the polysemous word were extracted. Most of them use the VSM, although other models based on frequency are also introduced.

Frequency-based features where the EKS is a dictionary

These features are computed by comparing the words co-occurring with the polysemous word with the dictionary definitions of its possible senses.

- *Overlaps*: One of the first attempts to use dictionaries to disambiguate is that presented by Lesk [1986]. In this work, the context was a window of some words surrounding the ambiguous word, for which the possible senses were collected from a dictionary. The dictionary definitions of all its possible senses were contrasted with the words surrounding the target word, and the frequency metric was as simple as counting overlaps: whenever a common word was encountered both in the dictionary definition for a certain sense and in

the context, a counter for that sense was increased. The selected sense is the one receiving the maximum score.

Lesk performed tests using four different dictionaries[4] to extract the definitions. The author points out that one of the keys for a successful disambiguation was the amount of information contained in each one of the dictionary entries: the best results came along with those dictionaries that have a longer definition. Intuitively, the longer the definition, the more probable will be finding an overlap with the context's words.

- *Box codes and Subject codes*: Some methods used the LDOCE dictionary classification of "box codes" and "subject codes" to refine the disambiguation results. This dictionary presents for each possible sense, a box code and a subject code. The box code specifies a sense's primitive (abstract, animate, human, etc.) and the subject code classifies the senses of a word by its domain, or "subject" (economics, engineering, etc.).

  [Guthrie et al., 1991] use the subject codes in the LDOCE dictionary in their 2-steps disambiguation method: first, the domain of the ambiguous word is selected and, once the domain is selected, a second step decides which of the senses pertaining to that domain is the right one.

  The selection of the domain is carried out by means of the subject codes in the dictionary, and the results of pre-processing the dictionary to find out, for each "subject code" (a domain category), the 20 most related words to the code as per the frequency of their occurrence is definition texts corresponding to the category in question. This first step ends when, using the context of the ambiguous word, the overlaps of context words with words in the "most related words" groups determine the domain to be selected.

  Then, a second step checks the senses of the ambiguous word that fall into that domain category (subject code), usually only one or just a few, and using a Lesk-based algorithm, the definitions of those senses are compared with the context to select the sense by counting, again, overlaps of words.

- *Frequency using the Vector Space Model*: Wilks et al. [1990] presents an early implementation of what would later be known as the VSM approach. In this case, a dictionary, the LDOCE, is used to extract both the short sentences that are used as context, as the senses, which are extracted from the dictionary definitions. The polysemous word's context and the senses definitions are transformed into vectors. In Wilks' approach each dimension is a word from the controlled vocabulary of LDOCE. Then, for the context words, the weights assigned to each dimension are a function of the co-occurrence of the words in the context with the words in the controlled vocabulary, computed throughout the whole dictionary. The same procedure is used to build the vectors for the candidate senses: the weights are a function of the co-occurrences of the words in the sense definition with the words in the controlled vocabulary. Wilks used seven weight functions in his tests, finding that amongst

[4](1) Oxford Advanced Learner's Dictionary of Current English (2) Merriam-Webster 7th New Collegiate (3) Collins English Dictionary (4) Oxford English Dictionary

them the best results came up with raw co-occurrence[5]. He also tested different similarity metrics, finding that the cosine angle between vectors was one of the best.

Frequency-based features where the EKS is a thesaurus

- *Category connected words*: Some techniques try to enhance disambiguation results by identifying, for each of the thesaurus categories, the set of words which are found to co-occur frequently in texts with the words representing the category in the thesaurus. [Yarowsky, 1992] uses the Roget's Thesaurus and the Grolier's Encyclopedia as external sources of information. Before the disambiguation, for each thesaurus class, a group of "salient" words that are representative of that class are selected. This selection is done by searching texts extracted from the encyclopedia that are representative of each class. To extract these representative texts for a particular class, the words in the thesaurus class are searched in the encyclopedia and a fixed-size context is extracted. The salient words in the contexts are selected using a weight based on a mutual-information-like estimate:

$$\log \frac{Pr(w|RCat)}{Pr(w)} \tag{3.9}$$

that is, *log* probability of a word *w* appearing in the context of a Roget category divided by the overall probability in the corpus. Once these groups of salient words have been assigned to each category in the thesaurus, the disambiguation is performed using a 100-word context for each polysemous word and a Bayesian classifier. In this work "to disambiguate" means to assign a category in the thesaurus, not a particular sense, to the polysemous word. Nevertheless, the authors point a way to map this output to other sense representations.

Frequency-based features where the EKS is an annotated corpus

The values of these features are computed from annotated corpora to extract information about the frequencies of the words contained in them. The words in the corpora have a tag that marks which sense is being used. Generally, the disambiguation process consists in two steps: one step is dedicated to processing the annotated corpus to extract statistics, the *learning step* or *training step*, and a second step consists in applying the statistics to disambiguate free text. This is why these techniques (and also those that learn complete rules, from corpora) are known as *supervised learning*.

The corpus can be used to calculate baselines with which we can compare other procedures. A very straightforward use of annotated corpus consists in extracting statistics from the words usages in the training phase and, afterwards, simply using the most frequently used sense observed during the training to assign one of the candidate senses during the disambiguation. These statistics can consider polysemous words as standing alone without regards to the context, and compute

[5]Later works in disambiguation using the VSM tested other weights, like tf·idf and χ^2 presented above, which returned better results than the metrics used by Wilks in this work.

the frequencies of all the appearing senses of each word, or mutual-information like statistics, like co-occurrence of words in the corpus pairwise calculated or the metric used in Equation 3.9.

Some common baselines are described by Miller et al. [1994]. The senses contained in WordNet are considered as possible senses, and the Brown Corpus is used to extract frequencies and decide the assignment for the disambiguation:

- Individual frequency: each open-class word (noun, verb, adjective or adverb) in a group of passages of the Brown Corpus is tagged with the appropriate sense in WordNet. Using these tags, the frequency of each WordNet sense appearing in the corpus is computed. To disambiguate, given a certain polysemous word in a text and its part of the speech, the most frequent sense for that part of the speech in the corpus is assigned. If there is a tie, the assignment is randomly made between the equally frequent senses.

- Co-occurrence frequency: Now the co-occurring frequencies of the words are computed pairwise, and the result is annotated in a matrix. When disambiguating, the context is kept in a bag of words. When a polysemous word is encountered, the matrix is consulted searching the co-occurrence frequency between each word in the context and each of the possible senses. If only one sense co-occurs with any word in the context, this sense is selected. If more than one sense co-occurs with the context, the most frequent sense among them is selected. Ties are broken at random.

- Random assignment: All the senses are considered as equal and the assignment is made at random.

  [Miller et al., 1994] found that using the most frequent sense induces a significant improvement of accuracy in disambiguation than the random assignment, but the use of co-occurrence gave similar results to the simple frequency rule.

On the corpus, it is possible to apply any Machine Learning procedure to learn to label words with their senses. Some common learning procedures are based on decision lists, decision trees, Support Vector Machines, Maximum Entropy models, etc. In particular, Bayesian classifiers work in the following way: they take each word in a context, and using the sense-tagged corpus for training, make an estimation of the probability to find the word under a certain sense, as well as the probability of that sense in the entire corpus. The former calculation is done for all the possible senses and all the words in the context, and finally the senses are scored using a certain metric, usually a log-likelihood. The best scored sense is selected.

3.1.4.4 Rules-based features

To generate these WSD features typically a sample is used, generally in the form of an annotated corpus or in the form of a collection of annotated samples, to learn supervisedly rules about the usage of words inside the knowledge base. The rules take into consideration lexical collocations or syntactic relations, among other features, that model the way a sense is usually used in a text, and that is distinguishable from the way other senses are used in a sentence. Hence, the context

surrounding the polysemous word has to keep the relations between its words, that is to say, a relational context as explained in Section 3.1.2 is used. These approaches use microcontext, usually one sentence or a few words long, to seek for relations between words.

Like the frequency-based methods, some rules-based methods take advantage of the statistical processing of corpora, but instead of searching frequencies of words or co-occurrences of words, they consider the frequency under which a certain *collocation*, or a combination of collocations, is observed in the corpus to decide if a rule can be generalised from a frequently observed usage or not.

- Simple collocation: [Yarowsky, 1993] test several syntactic and positional simple collocations in a WSD exercise. They use an unannotated corpus that mixes different text types: news, scientific abstracts, encyclopedic entries, etc. to minimise the effect of discourse style. From this corpus, samples of collocations of polysemous words are extracted, and the right sense is manually annotated. This study was centred in binary sense distinctions. The collocations studied were: first adjacent word to the right of the polysemous word, to the left, first right/left content word, subject/verb pair, verb/object pair and adjective/noun pair.

  Given a certain type of collocation and the senses of an ambiguous word, words participating in the collocation with the ambiguous target word are extracted from the samples, and the frequency for each sense is computed. The disambiguation can be done using only one type of collocation, or all of them. If only one type of collocation is in focus, each new appearance of an ambiguous word and a context word connected to it through the collocation is disambiguated by searching in the frequencies list the pair target word/context word, and assigning the sense that was more frequent in the samples. If all the collocations are to be considered for disambiguation, once an ambiguous word is encountered its context is analysed to check if any collocation rule is matched. If more than one collocation apply in a context, the one chosen is the one that is located first in a decision list.

- Multiple collocations: [Hearst, 1991] proposes a disambiguation system that combine multiple collocations and features to disambiguate. The features are learnt from a hand-tagged corpus (Grolier's Encyclopedia). The corpus is initially pre-processed with a PoS-tagger and fragmented into prepositional phrases, noun phrases and verbs groups. After that, a manually-defined list of properties and collocations is used to process the corpus and extract matching contexts (rules) for the polysemous words (this work considers five binary sense target words), computing the frequency observed for each sense that participates in a matching context. During the experimental phase, whenever one of the five polysemous words considered is found in the corpus, a certain metric, based on the frequency observed during the training phase for each feature participating in the context for that word, is used to assign the right sense.

- Information theory classifiers: Using bilingual corpora and a certain syntactic relation a word can be disambiguated inside a context. Those polysemous words that take a different

form per sense in other language can be successfully disambiguated clustering its usages in a second language corpus. For instance, in a machine-translation approach, the polysemous French verb "prendre" is translated in a English-French corpus as "take","make"," rise" and "speak". Using the syntactic relation *object*, the words "mesure", "note", "exemple", "décision", "parole" are extracted from the French corpus. Clustering the group of verb translations and the group of objects as per the frequencies of use, the results show that the sense "take" should be used when the object in French is "mesure, note, exemple" and the senses "make, rise, speak" should be used when the object in fresh is "decision, parole". The two groups of French objects correspond to two senses of the French verb [Brown et al., 1991].

3.1.5 Competitive evaluations

The development of WSD in the last few years has been strongly promoted by the three SENSEVAL evaluations[6] [Kilgarriff, 1998, Kilgarriff and Palmer, 2000, Edmonds and Cotton, 2001, Mihalcea et al., 2004a], and more recently the SEMEVAL-2007[7] workshop [Pradhan et al., 2007]. As mentioned before, the trend along the last year has been to combine different features, obtained mainly from annotated corpora, lexicosemantic networks, textual repositories and dictionaries, using Machine-Learning algorithms to train a classifier able to annotate the words with their appropriate sense.

In particular, the WSD exercise for English all words in SENSEVAL-3 consisted in disambiguating nouns, adjectives and verbs from a sample corpus extracted from the British National Corpus, and the selected sense inventories were WordNet (for nouns and adjectives) and WordSmyth[8] (for verbs). The best scored systems reported accuracies near 73% for fine sense distinctions [Mihalcea et al., 2004a].

In SEMEVAL-2007, in the task of disambiguating all words in English, the best reported results have an F-score of around 89% [Agirre and Lopez de Lacalle, 2007, Hawker, 2007, Cai et al., 2007] using the sense-annotated corpus from OntoNotes Hovy et al. [2006], and 59% [Tratz et al., 2007, Chan et al., 2007, Mihalcea et al., 2007] using WordNet senses. The reason for this lower score is that, given to the fine-grained senses of WordNet, the task was much harder, as shown by the fact that the inter-annotator agreement was only 72% for verbs and 86% for nouns. Another SEMEVAL competition is scheduled to take place in 2010.

3.1.6 Discussion

The first attempts to solve natural language ambiguities date from the fifties, and since then the approximations tested have been several. Some of the disambiguation lines, like the complex approaches from the time of the AI knowledge sources have been withdrawn, mainly due to the enormous manual effort that was required to maintain the knowledge base. On the opposite side,

[6] http://www.senseval.org/
[7] http://www.senseval.org/
[8] http://www.wordsmyth.net

the use of machine readable dictionaries and thesauri make available a vast amount of knowledge. Nevertheless, dictionary inconsistencies have to be taken into account as a limitation to the disambiguation performance. Another important conclusion is that encyclopedic knowledge, due to its larger amount of information, can perform better than brief dictionary definitions when using the frequency-based approaches [Lesk, 1986]. The most extensively used approaches are those based on computational lexicons, mainly due to their feature of compelling altogether dictionary glosses, a taxonomy and a network of relations, which opens the way to use them as a basis of most of the techniques presented for those EKS. Moreover, WordNet has been created by hand, which eliminates most of the errors cumulated when automatically processing natural language to extract the knowledge source, and has served as a basis for continuous enrichment through more or less automatised methods.

Unlike dictionaries, raw corpora do not indicate which sense of a word occurs at a given instance, so hand tagging of training examples led to annotated corpora. The use of annotated corpora has been the object of many works during the nineties, but data sparseness is a difficult problem that is not completely solved, and even using bootstrapping for semi automatic annotation, there is a significant human intervention required for each word in the vocabulary [Yarowsky, 1992]. Ide and Véronis [1998] believed that the limit of what could be done with a corpus was apparently reached by ends of the nineties, although most of the systems that participated in the last SENSEVAL and SEMEVAL competitions, in 2004 and 2007, used corpus-based Machine Learning methods. It is interesting to cite here the work by Fernández et al. [2004] on automatically collecting, from the Internet, contexts in which a word is used with a particular sense, which helped the construction of sense-tagged corpora.

Regarding the disambiguation approaches, the exploitation of the hierarchical structure of a taxonomy to expand the amount of background data to context words' ancestors and connected words is quite useful [Mihalcea et al., 2004b]. Frequency-based methods can take benefit of this fact, as seen in [Yarowsky, 1992, Resnik, 1995a], so a combination of both network-structure-based approaches and frequency-based approaches seems to be a good methodology. On the other hand, rules-based methods have not attracted much interest, probably due to the fact that the implemented approaches require an annotated corpus or a hand tagged collection of samples to learn the rules. Nevertheless, some authors [Wilks et al., 1990, Resnik, 1995a, Mihalcea et al., 2004b] mention that keeping more information from the context than a mere bag-of-words, such as syntax, collocations or other type of relations, should enhance their results.

Word Sense disambiguation has improved much in the last years, pushed particularly by the organization of the SENSEVAL and SEMEVAL competitions, although there is still much room for improvement. A problem of the first evaluations was the use of very fine-grained lexical repositories, with low inter-annotator agreement rates. Work in the area is underway to overcome these issues.

3.2 Information Extraction

Information Extraction (IE) consists in extracting organized information (entities, events and relationships) from unrestricted natural language text. An application example can be automatically identifying within news a particular event, like terrorist attacs, natural disasters or company merge information. IE may also deal with attributes linked to those events, for instance the date and location in which the events took place; the people, companies or other organizations involved; also money quantities transferred. IE has been defined traditionally as filling in templates or databases departing from a natural language textual sequence [Ranshaw and Weischedel, 2005].

There is a broad field of applicability for information extraction techniques, being useful whenever there is a need to structure information that is only available as natural language texts, like in the case of news services or in semantic information retrieval from texts. Currently, e-mail providers use IE techniques to identify events in their users' e-mails and propose their storage in the personal calendar, or to detect locations and people and suggest links to encyclopedias or online documents that expand the information about the detected entities. Another application consists in associating job offers and job seekers' profiles in jobs posting engines. There are also works in the fields of automatic generation of summaries, text mining and language understanding.

Currently the extraction of information is a mature research field, accumulating approximately two decades of investigation. At least two facts triggered this research line at the beginnings of the nineties: on the one hand, the improvement on computational and storage capabilities, which gave way to large quantities of textual data available in knowledge repositories, hard to be processed manually. On the other hand, an important impetus was provided by the ARPA-funded Message Understanding Conferences (MUCs), whose main contribution was to make available a common testing frame for the evaluation of different IE proposals through standardised data sets and evaluation metrics. The first conference, MUC-1, organised in 1987 was an exploratory approach without a formal evaluation proposal using texts about naval sights. MUC-2 formalised the task of filling templates, with ten slots per template, and the evaluation was done using precision and recall metrics described some time later [Grishman and Sundheim, 1996]. In the next two MUCs (1991 and 1992), the subject was Latin American terrorism, and the templates complexity increased so that the number of slots in MUC-4 was yet 24 per template. One of the main conclusions from these early conferences was that a system can attain a high accuracy without performing a deep linguistic analysis of the sentences, e.g. [Lehrnert et al., 1994] and [Cardie, 1993].

From MUC-5 on, the conference was conducted as part of the TIPSTER project. Topics changed again, including financial news, and texts in Japanese language were added. In MUC-6[9] and MUC-7[10] the tasks were:

- Named Entity Recognition and Classification: Identifying persons, organisations, currency, etc.

[9] http://cs.nyu.edu/cs/faculty/grisham/muc6.html

[10] http://www-nlpit.nist.gov/related_projects/muc/proceedings/muc_7_toc.html

- Coreference Resolution: Identifying which mentions refer to the same entity in the real world.
- Element Template: Filling templates about people and organisations.
- Relation Element: Filling templates that represent relationships between entities (located in, works for, produced by, etc).
- Scenario Template: Filling templates about events mentioned in the texts.

Four years after MUC-7 there have been two conferences organised within CoNLL (*Conferences on Computational Natural Language Learning* oriented to the recognition and classification of language-independent named entities[11]. Similarly, the NIST (*National Institute of Standards and Technology*) organises periodic competitions called *Automatic Content Extraction Evaluation* (ACE[12]) since 2001. This competition includes tasks for entity recognition, mentions, values, relations and events, as well as a task for the recognition and evaluation of temporal expressions.

The following subsection introduces the tasks of Entity Recognition and Coreference Resolution. Relationships Extraction will be treated separately in depth as it will be a main focus of this work.

3.2.1 Named Entities and Coreference Resolution

Named Entities Recognition and Classification (*NERC*) consists in, given a set of predefined categories, automatically locate in a text the words corresponding to those categories. As an illustration, if the categories under study are *person, organisation, location* and *date*, given the following text extracted from Wikipedia:

> Angela Dorothea Merkel, born in Hamburg, Germany, on July 17, 1954, as Angela Dorothea Kasner, is the Chancellor of Germany. Merkel, elected to the German Parliament from Mecklenburg-Western Pomerania, has been the chairwoman of the Christian Democratic Union (CDU) since April 9, 2000. She has been the chairwoman of the CDU-CSU parliamentary party group from 2002 to 2005. The most powerful woman in the world, as considered by the Forbes Magazine, is only the third woman to serve n the G8 after Margaret Thatcher of the UK and Kim Campbell of Canada.

the task consists in locating and annotating in the text the words corresponding to one of those categories:

> [person Angela Dorothea Merkel], born in [location Hamburg], [location Germany], on [date July 17, 1954], as [person Angela Dorothea Kasner], is the Chancellor of [location Germany]. [person Merkel], elected to the [organisation German Parliament] from [location Mecklenburg-Western Pomerania], has been the chairwoman of the [organisation Christian Democratic

[11] http://www.cnts.ua.ac.be/conll2002/ and http://www.cnts.ua.ac.be/conll2003/
[12] http://www.nist.gov/speech/tests/ace/

Union (CDU)] since [date April 9, 2000]. She has been the chairwoman of the [organisation CDU-CSU] parliamentary party group from [date 2002 to 2005]. The most powerful woman in [location the world], as considered by the [organisation Forbes Magazine], is only the third woman to serve in the [organisation G8] after [person Margaret Thatcher] of the [location UK] and [person Kim Campbell] of [location Canada].

In some cases, for the sake of simplicity, no annealing is permitted within entities (e.g. [organisation Patents Office of [location Bern]]). This allows to approximate the Entity Recognition problem as a word-based classification and tagging task, where the words or words sequences are tagged as *person, organisation, location, date* or *non-entity*, and there is no need to use more complex annotations like analysis trees.

The evaluation is usually given in terms of *precision* and *recall*. *Precision* is the percent of located entities that are right, and *recall* is the percent of entities in the text that the system was able to recognise. These two metrics may be combined in a single metric, called *F-score*, defined as the harmonic average between precision and recall, $\frac{2PC}{P+C}$.

The most common techniques used for NERC in the MUC tasks can be classified in three types:

- Knowledge-based systems, which use rules, patterns or grammars, many times hand-crafted but also derived using heuristics from annotated corpora (e.g. Riloff [1996], Soderland [1999], Arevalo et al. [2004]. For example, given the pattern:

  ```
  Xxxxx+ is the PROF
  ```

  Xxxx+ represents a words sequence starting with a capital letter and PROF is a profession name. The patterns is deemed to carry enough evidence to select the sequence of words as a persons' name. Similarly, persons' titles such as Mr., Mrs., Exc., etc., followed by word starting with capital letter do indicate the presence of the entity *person*. Geographical dictionaries (*gazetteers*) and exhaustive lists of name-surname pairs are also common in these systems. There are languages for patterns, such as *JAPE* [Cunningham et al., 2002], created to facilitate the task of hand-writing rules and the analysers for entity recognition. Hand-crafted patterns usually attain a high precision but the recall is low,since it is hard to foresee all the natural language contexts in which an entity can be used.

- Systems based on automatic learning models, including memory-based learning, maximum entropy and hidden Markov models [Freitag and McCallum, 2000, Klein et al., 2003, Florian et al., 2003, Kozareva et al., 2005], transformation lists [Black and Vasilakopoulos, 2002], boosting algorithms [Carreras et al., 2003] and support vector machines [Isozaki and Kazawa, 2002, Mayfield et al., 2003, Li et al., 2005b].

- Systems that combine knowledge-based and automatic learning techniques, for instance applying each of them at different steps during the recognition process [Mikheev et al., 1998, 1999].

In MUC-6 and MUC-7 the Named Entity Recognition task already included the following categories: person, organisation, location, date, time, money and percents. In both competitions

the best systems for English language NERC (e.g. [Fischer et al., 1995, Gaizauskas et al., 1995, Mikheev et al., 1998]) attained F scores around 90%. In particular the latest attained a precision of 95% and recall of 92%, an overall F score of 93.39%, quite near the results obtained by human annotators in the same texts (F scores of 96.95% and 97.6%). Due to the very good results obtained in these conferences, later competitions followed the trend to generalise the Named Entity Recognition task, trying to obtain systems with higher portability to different languages and domains, and systems with a higher integration of the entity recognition and the coreference resolution.

3.2.1.1 Development of systems with enhanced portability among languages

Yet at early MUC conferences there were tasks for NER in Chinese and Japanese, within the so-called *Multilingual Entity Task Evaluation (MET-2)*[13]. More recently, two CoNLL conferences have been organized including Dutch, German, English and Spanish (CoNLL.2002 and 2003), and four entity categories: person, location, organisation and miscellaneous entity (other entity types grouped into a single category). It must be noted also that these competitions consolidated the systems based on automatic learning algorithms using the annotated corpora provided.

As an example, Table 3.1 shows the initial sentence in a text about Albert Einstein, whose annotations follow guidelines similar to CoNLL-2002 and CoNLL-2003. The first column contains the words int he text, the second contains the part-of-the-speech (e.g. NN:noun, NNP: proper name, VBP:verb participle, IN:preposition, CD:cardinal number, comma:punctuation sign). The third column shows phrasal information using IOB annotation: words pertaining to a noun phrase are marked I-NP, or B-NP when preceded by other noun phrase; conversely, I-VP and B-VP is used for verb phrases. Words outside these phrasal types are marked as O. The last column shows the words annotated with their entity type, which the system must learn to identify. In the example, I-PER refers to persons, I-LOC to locations and I-MISC are miscellaneous entities such as dates.

3.2.1.2 Extension of the categories to conceptual hierarchies and enhanced portability among application domains

MUC-7 already included a basic sub-classification of entities, for instance organisations where sub-classified into governmental, private companies and other; persons where sub-classified into military or civilian and other; locations could be divided into cities, provinces, countries, regions, airports or body of water (lake, river, sea...). The subsequent ACE competitions used a more detailed classification into subtypes, for instance, Table 3.2 shows the entities types and subtypes included in ACE-2005 and ACE-2007.

The approaches to identify the type-subtype classification ranged from trying the direct classification into subtypes, to departing from the main types and afterwards stepwise refining each words group towards the subtypes. Some authors point out that this later approach would end up

[13] http://nlp.cs.nyu.edu/met2j/

Albert	NNP	I-NP	I-PER
Einstein	NNP	I-NP	I-PER
(	(	O	O
14	CD	I-NP	I-MISC
March	NN	I-NP	I-MISC
1879	CD	I-NP	I-MISC
-	-	O	O
18	CD	I-NP	I-MISC
April	NN	I-NP	I-MISC
1955	CD	I-NP	I-MISC
)	)	O	O
,	,	,	O
born	VBP	O	O
in	IN	I-PP	O
Germany	NNP	I-NP	I-LOC

Table 3.1: Example of annotation similar to CoNLL-2002/2003

Type	**Subtype**
Facility	Airport, Building-grounds, Path, Plant, Subarea-facility
Geo Political Entity	Continent, Country or District, GPE-cluster, Nation, Population center, Special,State or Province
Location	Address, Boundary, Celestial, Land region-natural, region-general, region-international, water body
Organization	Commercial, Educational, Entertainment, Government, Media, Medical Science, Non-governmental, Religious, Sports
Person	Group, Indeterminate, Individual
Vehicle	Air, Land, Subarea vehicle, Underspecified, Water
Weapon	Biological, Blunt, Chemical, Exploding, Nuclear

Table 3.2: Entity types and subtypes in ACE 2005 and 2007

in more exact results [Florian et al., 2006]. A particular case of a type-subtype hierarchy takes the form of an ontology. Following the line of named entity recognition, some authors deal with ontology population, focusing on searching named entities with the role of ontology instances in the texts, in order to link them with their corresponding concepts inside a hierarchy. Beyond the approach of using annotated corpora similar to those in the MUC and ACEs competitions, some methods try to reach that goal using unannotated texts [Alfonseca and Manandhar, 2002c, Cimiano and Volker, 2005, Sekine, 2006]. This task overlaps with the efforts on learning taxonomic relationships between concepts, which will be studied with more depth in following sections.

3.2.1.3 Integration of named entity recognition and coreference resolution

Coreference resolution consists in the location of all the expressions in a text that share a common referent in the real world. For instance, the highlighted expressions in the following text:

Albert Einstein (14 March 1879 - 18 April 1955), born in Germany, was **the most well-known and important scientist in the 20th century**. In 1905, while

> still being **a young and unknown scientist** working at the patents office in Bern (Switzerland), published his Theory of the Relativity. (...) **Einstein** was born in Ulm (Germany), 100 Km east from Stuttgart, in a Jewish family. **His** parents were Hermann Einstein and Pauline Koch. **His** father was a salesman that later on worked in the electrochemical industry.

correspond to different mentions of the referent *Albert Einstein*: two refer to his proper name, two are nominal expressions (*the most well-known and important scientist in the 20th century* and *a young and unknown scientist* and the rest are possessive determiners (*his*).

ACE competition considers that an entity include all the textual mentions including *named mentions* (e.g. "Albert Einstein"), *nominal mentions* (e.g. "the scientist") or *pronominal mentions* (e.g. "he", "his"...). This task if also referred to as *mention detection*. The current definition in ACE is more focused to detecting mentions within individual documents, although the trend of solving multi-document coreference in a whole corpus is gaining force [Daumé III and Marcu, 2005, Nicolae and Nicolae, 2006].

Some of the proposed solutions to this task are as follows:

- Characterising each proper name with a vector space model and using diverse functions to measure the relevance of conceptual terms (tf-idf, log-likelihood, etc). This is used to cluster all the mentions to the same name across different documents, grouping those references located in similar contexts [Bagga and Baldwin, 1998, Gooi and Allan, 2004, Kulkarni and Pedersen, 2005].

- Characterising each proper name with attributes and clustering the mentions that share common attributes [Mann and Yarowsky, 2003, Kalashnikov et al., 2007]. For example, it is sometimes possible to extract people's gender from named mentions preceded by "Mr." or "Mrs.", or when the given name clearly indicates it. It is also possible to use relations extraction techniques (see next section) to identify attributes such as people's birth date and place, allowing to discard that two mentions to the same proper name have the same referent in the real world when their birth dates are different.

- Training a machine learning algorithm with features found in the texts to classify them into coreferent and non-coreferent [Li et al., 2005a]. Possible features for the training are the context, the edit distance between the different mentions, whether the mentions contain exactly the same words (e.g. same name or same surname), whether a mention is another's mention acronym or whether the adjacent words to the mention are compatible (e.g. two mentions to the same referent can not start both with "Mr." and "Mrs."), among other. Bunescu and Pasca [2006] use attributes obtained from Wikipedia to train a support vector machine.

Following this trend, one of the tasks in the SEMEVAL competition consists in, given a web search and a group of URLs, grouping the resulting pages that refer to the same entity in the real world[14] [Artiles et al., 2005].

[14] http://nlp.uned.es/weps/

3.3 Relationships extraction

3.3.1 Introduction

We can define a relation as a triple (predicate, subject, object), which indicates some criterion according to which the subject and the object are connected. For example, the relationship *place of birth* connects instances of "Person" with the instances of "Place" corresponding to the places where they were born.

The need for wide-coverage knowledge bases containing a large number of entities and relations has always faced the so called *knowledge acquisition bottleneck*, i.e., the difficulty to acquire and model general knowledge from texts, or relevant knowledge to a particular domain [Hearst, 1992, Gale et al., 1993]. Current computer infrastructure is able to process much vaster amounts of information than what can be generated manually in a cost-effective way. Therefore, automating or semi-automating the acquisition of new knowledge from text is at the moment a very active area, which includes the automation of extraction of relationships amongst the concepts and instances existing in the lexicons.

Remember from Section 2.1 that the triples where the object is not a concept or an instance, but a scalar number, are commonly called *attributes*. This section will not make a distinction between attributes and pure relations, as many techniques are equally applicable to both cases. For example, similar patterns can be used to learn the relationship between a country and its capital city, and to learn the attribute *area* of a country.

Regarding the type of relationships to be extracted, the following distinction can be done:

- Extraction of taxonomic relations, i.e., IS-A relations that build up a taxonomy from the most general term in our domain, the *root* of the taxonomy, to the most specific concepts at the bottom of the hierarchy.
- Extraction of non-taxonomic relations, i.e., any link between two terms included in the network which does not imply a subsumption relation.

With reference to the methods that extract IS-A relations between concepts, it is important to point out that the task of finding taxonomic relations for new terms nonexistent in the base ontology is essentially equal to classifying these new terms inside the hierarchy, as many approaches mainly use the subsumption relation to position a term inside an ontology. Therefore, most of the approaches for the discovery of IS-A relations explained below, in practise, classify new terms inside an ontology.

3.3.2 A Classification of Techniques for RE

To extract taxonomic and non-taxonomic relations, several methods have been used. They may be classified as follows:

- Methods based on dictionaries

- Methods based on distributional properties of words
- Methods based on patterns
- Methods based on annotated corpora

3.3.2.1 Methods based on dictionaries

Dictionary definitions, although written in unrestricted text for human consumption, usually follow some stylistic guidelines that make them all have a similar structure, making them easier to process. Furthermore, they contain the most salient and relevant information about the concept being defined, so it is not necessary to prune the texts in order to discard irrelevant data. Ide and Véronis [1994] contain a detailed study of the advantages and limitations of the first approaches to use these resources.

One of the earliest references to the processing of dictionaries to extract relations can be found in Wilks et al. [1990], who introduce three different approaches to process the LDOCE dictionary in order to transform its contents into machine-tractable information structured as a network, dealing with disambiguation, parsing and classification issues. Although this work is not particularly focused in relationships extraction, the authors outline a method for the extraction of semantic relations amongst the concepts. To do so, they depart from dictionary definitions that have been already processed to create the concepts network, so the words in the definitions are already disambiguated. Then, manually defined case frames are applied to extract hyponymy and meronymy relations from the definitions. The hyponymy relation is extracted from the *genus word*[15], and the meronymy relation is identified searching lexical clues as "has the part", combined with proper predicate cases in the lexico-syntactic rule.

Rigau [1998] describes a system that analyses dictionary definitions to learn a lexico-semantic network similar to WordNet. His approach consists of identifying in the definitions the genus word and, next, disambiguating it, in order to structure the concepts according to the hypernymy relationship. An important contribution of this work is the set of heuristics and techniques used for the disambiguation of the genus word, and the identification of several of the problems found when extracting ontologies form dictionaries.

[Richardson et al., 1998] present a methodology to extract relations from dictionary definitions using a frequency-based approach. The authors use LDOCE and the American Heritage Dictionary to acquire a knowledge data base from the knowledge contained in the dictionaries. To do so, the concepts in the dictionary are represented in a network and linked by relations. Regarding the extraction of relationships, they use a syntax parser (the same used for the MS Word 97 grammar checker) and lexico-syntactic rules to extract a given set of relations from the dictionary entries. The rules are defined to extract attributes like *colour*, *size*, *time*, and relations between words like *cause*, *goal*, *part*, *possessor*, amongst other. One interesting asset of this approach is

[15]The genus word in a dictionary definition is the word identified for each entry by the pattern "a *definition entry* is a *genus word*", e.g., "a *dog* is a *mammal*"

that the learnt relations are extended to "similar" words, i.e., to words that are found similar to the original words for which the relation was found. To find the similar words, the authors use weighted paths: the paths in the network receive a weight based on the observed frequencies of the relations that they represent. Using those relations that showed a highest weight for a given word, similar words are found using patterns that combine the heavy-weighted relations. The similarity patterns were learnt in a training phase from a thesaurus. Then, the similar words receive the same relations hold by the original words. for instance, the relation watch/byMeans/telescope was not present originally in the network, but was inferred from observe/byMeans/telescope through the relation watch/isHyponym/observe.

Several other works use WordNet glosses to extract relationships:

- [Harabagiu and Moldovan, 1998, Harabagiu et al., 1999] present a work focused on the enrichment of WordNet where, amongst other tasks, a disambiguation of the WordNet's glosses is addressed. Harabagiu et al. propose parsing the disambiguated glosses and its annotation with the corresponding syntactic roles of the words that compose the gloss. Then, from the syntactic relations, some semantic relations can be extracted. For instance, depending on some defined constraints, different relations can be added using the subject-verb relation, e.g., for those verbs that are hyponyms of the verb *to cause*, the subject can be linked to the verb through the relation *agentOf*, e.g. from the gloss "a game played by two players", the relation player/agentOf/play is extracted. Similarly, some relations can be extracted from prepositional predicates, e.g., the prepositional phrase "sacrament of penance" found in the gloss for *confession* indicates a semantic KIND-OF relation: penance/isKindOf/sacrament. From adjectival or adverbial adjuncts, some attributes can be recognised, e.g., from "two players" the relation player/attribute/two can be derived.

- Dictionary glosses usually present a particular structure that can be exploited to extract relations by means of patterns. [Novischi, 2002] uses lexical patterns extracted from the WordNet glosses to disambiguate the words in a corpus. These patterns are extracted by identifying repetitive expressions used in the definitions which, by themselves, constitute indicators of certain relationships. Therefore, some relations can be extracted by searching typical constructs that introduce a gloss, for instance, expressions like "relating to", "genus of", "used in", "consisting of", amongst others.

- [DeBoni and Manandhar, 2002] use a pattern-based method to extract telic relations, that express the purpose or function of a concept. For instance *an agent HAS THE PURPOSE OF performing a certain action*, or *a medical diagnostic procedure HAS THE PURPOSE OF detecting diseases*. The authors extract this type of relationships using the WordNet glosses with the goal of enriching WordNet itself, in a two-steps procedure:

  - First, they process the glosses from set of synsets in the search of matchings to a group of manually defined patterns that model the use of the telic relation. The patterns are enhanced with part-of-speech. The results after this step are, for each one of

the processed WordNet entries, a word or group of words that are supposed to be connected to the entry through the telic relation. Some of the patterns (PoS omitted in the examples) and the extracted words are:

* "...to TARGET WORDS by the use of ...". This pattern, applied to the gloss of *mammography*, which is "a diagnostic procedure to detect breast tumours by the use of X rays" extracts "to detect breast tumours". Therefore, the relation extracted is *mammography/hasPurpose/to detect breast tumours*.
* "...used for TARGET WORDS". This pattern extracts, for instance, *tracing paper/hasPurpose/to trace drawings* from one of the processed glosses.
* "...used in TARGET WORDS". This pattern extracts, for example, the following telic relations for the concept "seal oil": *seal oil/hasPurpose/making soap*, *seal oil/hasPurpose/dressing leather*, *seal oil/hasPurpose/lubrification*.

– In the second step, the word or group of words extracted in the previous step are disambiguated in order to attach them to one WordNet synset, the one that represents the concept expressed in the word or group of words. This way, the telic relation is defined between two WordNet synsets and can be included in the lexicon without ambiguities.

The authors reported an accuracy of 78% in the extraction of relations, 77% in the disambiguation step, and a 60% total accuracy.

The availability of the disambiguated glosses for the English WordNet[16] makes it possible to extract relationships from the WordNet definitions where the object of the relation is already disambiguated as other WordNet synset, so the obtained relationship is automatically ontologised without the need of using a WSD method.

3.3.2.2 Methods based on distributional properties of words

These systems rely on the distributional hypothesis, i.e., that semantically similar terms share similar contexts, and therefore the co-occurrence distributions of terms, as well as the distribution of collocations, can be used to calculate a certain *semantic distance* between the concepts represented by those terms. Using this distance, a partial ordering between the terms can be calculated and thus a taxonomy may be obtained, or the terms can be classified inside an existing taxonomy. These statistical methods are mainly used to extract taxonomic relations, although they also have application for extracting non-taxonomic relations [Maedche and Staab, 2000]. We can group these methods in two categories:

- Clustering methods to derive taxonomic relations
- Statistical methods for the discovery of non-taxonomic relations

[16] http://wordnet.princeton.edu/glosstag

Clustering methods to derive taxonomic relations As far as the hyponymy-hypernymy relation is the basis for the construction of a hierarchy of concepts, most of the methods developed to automatically build a taxonomy through conceptual clustering address the problem of defining subsumption relations between terms. From the Machine Learning field, the clustering of terms in a hierarchy consists in, given a set of totally or partially ungrouped concepts, group incrementally the semantically similar terms in clusters and order the clusters in a hierarchy using a taxonomic relation. The clustering task can be performed under different approaches [Steinbach et al., 2000], some of which are presented next:

[Maedche and Staab, 2001, Faure and Nédellec, 1998, Caraballo, 1999] use a bottom-up clustering algorithm, also called *agglomerative clustering*, to build a hierarchy of concepts. It is a procedure that starts considering the set of individual concepts as the set of initial clusters, each cluster containing one single concept. In successive steps, a pair of clusters is merged at each step according to the similarity found between the terms in each cluster. The similarity between clusters can be computed using any of the several VSM-based similarity metrics already introduced in Section 3.1.4.3. The criteria for merging are various. Some of the most well-known are the two clusters with the two more similar concepts, and the two clusters so that the averages of their elements is the most similar, amongst other [Matsumoto, 2003].

An example of this approach is the ASIUM tool [Faure and Nédellec, 1998]. It applies a bottom-up clustering algorithm to learn a hierarchy of nouns as per the verbs with which these nouns are syntactically related.

They first extract the concepts from the texts. To do so, they use *categorisation frames* that take the following form:

(verb) (syntactic role OR preposition:noun* OR concept*)*

E.g.: (to travel)(subject:human)(by:vehicle)

The subcategorising frames can be manually defined or learnt from natural language texts. The learning is done by extracting from the text some frames containing instances (in the above example: (to travel)(subject: my father)(by:car)). When a frame is frequently observed for a group of instances, the assumption that these instances represent the same concept is taken. For instance, if several sentences contain the verb (to travel) with the preposition “by”, and after the preposition appear car, train and motor-cycle, all the later terms are assigned to a same cluster and a concept is created for them: vehicle. This way the concepts are learnt from text.

The next step is to order the learnt concepts in a hierarchy. To carry out this task, the clusters of words are compared pairwise measuring the semantic distance between clusters according to the frequencies of their instances, e.g.:

verb: to cook in	verb: to put in
oven (4)	oven (5)
stew pan (12)	stew pan (3)
frying pan (12)	wok (6)
	pan (2)

Observed frequencies in parenthesis.

If the categorisation frames inside two clusters contain the same words with the same frequencies, they represent the same concept (distance = 0). If there are differences in the frequencies or the words, they are not same but (more or less) similar ($0 <$ distance < 1). Two clusters are completely disjoint when they do not share common instances (distance = 1).

The hierarchy is built from down to top: the lowest level is formed by the individual clusters. In the next level, given a similarity threshold, the similar clusters are merged. The merged clusters correspond to a higher level in the hierarchy, where the formed concepts have a more general meaning that subsumes the meaning of its forming clusters. The semantic distance is re-calculated for all the clusters and the merge procedure is repeated for the higher levels, up to the top of the hierarchy.

[Pereira et al., 1999] use a top-down clustering algorithm, where, in a similar way to bottom-up, the hierarchy may be obtained from the root and a similarity metric that measures *divergence* is used to assign the members of a cluster.

Some approaches mix the pure statistic clustering techniques explained above with other techniques:

[Cimiano et al., 2004b], use a clustering technique based on Formal Concept Analysis (FCA) [Ganter and Wille, 1999]. FCA is a representation formalism under which a data set can be modelled through different attributes. FCA *scaling* techniques process data through logical rules of features, in such a way that it is possible to arrange concepts in a *lattice* that represents how these concepts share certain values for the features, the attributes. From the lattice, a partial order between the concepts can be derived, and thus a hierarchy is defined.

[Caraballo, 1999] uses a bottom-up clustering algorithm enhanced with a pattern-based method to discover hyponymy relations (explained below). One of the problems of the clustering algorithms is that after the statistical ordering the resulting hierarchy is usually unlabelled. There is a need to assign a name to the concept represented by all the instances in the cluster. Caraballo uses the lexico syntactic patterns defined by [Hearst, 1992] to detect the hyperonyms of each cluster and assign a label to it.

Statistical methods for the discovery of non-taxonomic relations [Maedche and Staab, 2000, Kietz et al., 2000] discover non-taxonomic relations from texts using a corpus (unannotated), a lexicon, a predefined domain taxonomy and some language processing tools.

The first stage of their methodology consists in a linguistic processing of the corpus. In their experiments, an on-line corpus downloaded from a tourist information web is processed with a language processor that includes: tokenisation, identification of multi-word expressions and identification of specific words about the tourism domain.The system next elicits the concepts represented by those instances, annotates the part-of-speech, chunks the texts into phrases and annotates syntax. Several *dependency relations* are defined from the syntactic relations found between concepts, and are augmented through some heuristics (e.g. attaching prepositional phrases

to adjacent noun phrases), so at the end of the linguistic processing step the dependency relations are used to extract a set of pairwise connected words, relevant to the domain under consideration.

In a second stage, the set of pairs of words are analysed to find out non taxonomic relations. The domain-specific input taxonomy is used to augment each pair with its corresponding ancestors. The corpus is re-processed to extract the frequencies of individual domain words and co-occurrence of each pair of words, including not only the connected words found in the previous stage, but also computing co-occurrences between their ancestors. The co-occurring pairs that include two taxonomically related words are discarded, as only non taxonomic relations are in focus. Also, whenever there is a pair of words such that one ancestor of each is in another extracted pair, the first one is pruned out. Thanks to this pruning the relation is established between the concepts in the highest level of the taxonomy. For instance, a relation between "area" and "accommodation" is established rather than between "area" and "hotel", or a relation between "room" and "furniture" is preferred than the relation between "room" and "T.V.". The frequencies extracted from the corpus are used to compute *support* and *confidence* metrics, a balance of which, imposing a threshold, is used to decide which pairs of concepts will hold a (non-taxonomic) relation between them. The authors of this work do not address the problem of how to label the relations found by using their procedure. Nevertheless, they point out that, given that the error rate of this approach is about 33% for the best empirical setting, this method is intended to be used integrated with an ontology editor, as a help to the ontologist. Under this point of view, the labels of the relation should be defined in a manual way, by the ontologist.

3.3.2.3 Methods based on pattern extraction and matching

Most of these systems rely on lexical or lexico-semantic patterns to discover taxonomic and non-taxonomic relationships between concepts. Apart from a few exceptions (see [Navigli and Velardi, 2003] below), these systems use lexical and/or syntactic sequences to search a free-text corpus. Whenever a match is found, the words participating in the pattern are considered candidates for holding a relationship. In most of the systems the patterns are manually defined. We may differentiate among:

- Patterns for the extraction of taxonomic relations
- Patterns for the extraction of non taxonomic relations
- Patterns that learn relations from search data

Patterns for the extraction of taxonomic relations One of the first and most referred works are those of Marti Hearst [Hearst, 1992, 1998], who defines regular expressions to automatically extract hyponymy relations from unrestricted text. Hearst looks for pairs of words related in WordNet and extracts sentences where the words co-occur from a corpus. By finding commonalities among the contexts of the words, the patterns are manually built. Some of the patterns presented by the author are:

`such NP as NP*`

The asterisk indicates that a sequence of items of the indicated type (in this case, noun-phrases) is allowed in that position of the pattern.

Applying the above pattern to a free-text corpus, sentences like *"such authors as Herrick, Goldsmith,..."* would indicate that *Herrick* is a hyponym of *author*, *Goldsmith* is a hyponym of *author*, etc.

`NP , NP*, or other NP`

Extracts hyponymy relations (bruise/broken bone IS-A injury) from sentences like *"Bruises, ..., broken bones or other injuries*.

`NP , especially NP,* or|and NP`

This pattern would extract the hyponymy relations (France/England/Spain IS-A European country) from sentences like *"...most European countries, especially France, England, and Spain"*.

In her works, Hearst applies some linguistic tools, a part-of-speech tagger and a noun-phrase recogniser, to label the text used to extract the relations. Hearst uses texts from different sources, like the Grolier's encyclopedia or The New York Times newspaper. The relations discovered in these texts by her patterns are afterwards used to augment the WordNet lexicon.

Kietz et al. [2000] present a semi-automatic ontology acquisition approach that uses a corpora intranet of an insurance company as free text source of knowledge. The ontology learning tool works jointly with natural language processing tools, using different information resources like a lexical data base (for the linguistic processing), a core ontology (GermaNet, a German version of WordNet), and several general and domain-specific free texts used basically for two tasks: pruning the core ontology to a domain-specific ontology and extracting information for the enrichment of the domain-specific ontology.

Regarding the enrichment of the ontology with taxonomic relationships, the authors use a similar approach to that of Hearst [1992], combined with heuristics: (1) Compound words in German can be a source of hyponymy relations. For instance, *Arbeitlosenentschädigung* (unemployment benefits) is decomposed and the whole compound is suggested as hyponym of the last part of the compound (*unemployment benefit* IS-A *benefit*). (2) The same idea can be applied to phrasal compounds, e.g., *Automatic Debit Transfer* IS-A *Transfer*. The total accuracy of this method resulted in 68%.

The extraction of non-taxonomic relations is done using the statistical approach reviewed above Maedche and Staab [2000].

Alfonseca and Manandhar [2002b] depict a combination of pattern-based and a distributional-based approach, also for hypernymy. In a similar way to the procedure of Hearst [1998], WordNet is used to extract pairs of hyperonym-hyponym synsets. Instead of using an encyclopedic or a news corpus, the authors use the World Wide Web as free text. Using the Altavista search engine, the method retrieves web pages querying the words in the synsets of the related pairs. These pages are processed up to the syntactic level using linguistic tools. Then, the pages are parsed to extract

the sentences that contain both any of the hyperonym synset words and any of the hyponym synset words, and the syntactic dependencies between the hyponym-hyperonym pair are considered as a pattern. Some examples of the rules presented are:

From the sentence *Shakespeare was a first class poet* the following pattern can be extracted:

```
N1(subject) to be N2(object)
```

indicating that N1 IS-A N2

From *Shakespeare, the poet, ...*:

```
N1 (appositive) N2
```

indicating that N1 IS-A N2

In this method, those patterns for which a low frequency is observed are pruned. Some of the advantages of this method are that the patterns are automatically acquired, and that the data sparseness problem is lowered as the corpus used is as large as the WWW. On the other hand, most of the learnt rules were too general, for instance the second rule above would match to any appositive sequence of names. The patterns are used in this work to help a classification algorithm and they work well for this proposal, but this method should be revised, specially to enhance the generalisation of the patterns, if it has to be used in a standalone way for relationships extraction.

Cimiano et al. [2004a] presents a method that locates hypernymy (IS-A) relations using the WWW as corpus by means of hand-written patterns. The patterns defined are diverse: they use the Hearst patterns for IS-A relations, patterns created to find definites (noun phrases introduced by the definite determiner "the" and where the subject and object of the relation appear explicitly, e.g., *the Hilton hotel*), and appositional (*Excelsior, a hotel in...*) and copulative (*The Excelsior is a hotel in...*) sentences. The system locates proper nouns in web pages, then, each proper noun is aligned in the hand crafted patterns and a query to Google is performed in order to find the words that complete the patterns resulting in a match. The matching words are candidates to hold a hypernymy relation with the initial proper noun. The authors used different methods to weight the matchings (based on the frequency of occurrence of a same matching, or in human judgement), so not all the extracted candidates were considered, but only those that overcame a certain threshold. Several results are presented in the paper depending on the selected threshold, and for the best recall-precision setting (best F-score), the precision reached 39%.

Patterns for the extraction of non-taxonomic relations In a similar way to that of Hearst's, Berland and Charniak [1999] use manual defined patterns to find PART-OF-WHOLE relationships (holonymy relations). They use six seed words (book, building, car, hospital, plant and school) and five hand written lexical patterns that take into consideration the part of the speech of its components.

The patterns were applied to a News corpus (NANC: North American News Corpus) to create a group of selected words that match the patterns as candidate for a PART-OF relation. The set

of selected words was pruned by filtering those terms ending with -ing, -ness or -ity, suffixes that denote quality rather than a physical object. After that they test different metrics to find out which words seemed to hold the PART-OF relation in a stronger way: two metrics were tested, Dunning's log-likelihood or "surprise" metric, which returns a high value when the probability of finding a whole once the part is observed is much higher than the overall probability of finding the whole in the text, and Johnson's sigdiff (significant difference), which takes into account that, given a required threshold of probability, how far apart is the probability of finding the part and the probability of finding the whole once the part is observed. The second metric proved better. Taking into account the top 50 parts found for each of the 6 seed words, the overall precision of the experiment was 55%. If only the top 20 words are considered, this precision rises up to 70%.

Finkelstein-Landau and Morin [1999] learn patterns for the Merge relationship and for the Produce relation. The first relation is that of two companies merging into a new one, and the second is the relation that holds a company with its products.

The authors use a supervised and an unsupervised method to extract relations between terms. They define a method as "supervised" when the patterns consist in lexico-syntactic sequences manually defined, and the "unsupervised" method consists in defining manually some rules, like the type of terms that participate in a defined relation, how to recognise them, how many terms are required in the relation, cue words that indicate that the relation defined may be present in a context, etc.

As mentioned, the supervised method requires the predefinition of lexico-syntactic patterns, and the supervision of an expert. The method used is similar to that of Hearst [1992] . It consists in, for a certain type of relation (e.g. hypernymy), a list of pairs of terms linked by the relation has to be given (manually, or extracted from a thesaurus, or a database). Then, a corpus is crawled in the search of sentences where the related terms occur. The terms in each sentence are annotated with their PoS and syntactic role, and are generalised pairwise taking into account the longest common string that has the same syntactic function. Similar lexico-syntactic expression are clustered and each cluster is associated with a pattern. The patterns are validated by an expert and the accepted ones are applied to the corpus in order to extract new pairs of concepts which are candidates to fulfil the relation under study. The extracted pairs are again validated by the expert and reused as an input of the process, which is therefore enriched with its own results.

In the unsupervised method, once the goal is focused on a certain relation, e.g. merging (of companies), the terms that match the relation are automatically obtained from a corpus using predefined types of terms: in the merging relation, a triple is required (two companies and a word indicating the merging relation). Some key words are given for the particular relation under study: the company names are extracted by searching proper names containing substrings like "Ltd", "Corp" or "Inc", and the merging relation indicators are automatically extracted as words containing the substring "merge" (e.g. merger agreement, announce merger). For other relations, the extraction of terms takes into account syntactic and semantic relations. The relation is labelled using verbs or other words that appear to be frequent as part of the relation.

The supervised method for the Merge relation reported accuracies as high as 92%, 93% was

the accuracy for a combination of the supervised and the unsupervised method, and 72% was the accuracy for the unsupervised method. In the case of the Produce relation, the accuracy for the supervised method was 79%. The authors point out that the anaphora resolution would improve very much the results.

Navigli and Velardi [2003, 2004] describe an interesting approach to the extraction of non-taxonomic relations, what they call *semantic relations*, which is based on non-lexical patterns. In this case, the patterns take the form of rules over the concepts participating in the relation, and its ancestors, in such a way that each rule characterises the family of concepts that can participate in a given relationship. This task is integrated in a whole framework which, using general and domain-specific texts, a core ontology (e.g. WordNet), an annotated corpus, SemCor Miller et al. [1993], and some linguistic tools, performs terminology extraction and ontology construction. Basically, after applying a syntactic parser, the system extracts from a domain corpus a set of complex expressions (compounds -*credit card*-, adjective-nouns -*public transport*- and noun phrases -*board of directors*), selecting them through their different occurrence frequency in the domain texts vs. the general texts. In general, each complex expression is formed by two components (e.g. credit card, public transport, board of directors), a head (*card, transport, board*) and a modifier (*credit, public, director*). The component words are disambiguated assigning each one of them to a WordNet synset. The complex expressions are ordered in a taxonomy using the hyponymy-hypernymy relations found in WordNet for its head concepts. Then, the extraction of non-taxonomic relations is performed: the first step is to decide which relations are interesting for the domain under consideration. For instance, for a tourism information ontology, some of the semantic relations considered by the authors are the following: PLACE (the one that holds between hall/service in "the service has place in the hall"), TIME (afternoon/tea e.g., in "afternoon tea"), THEME (e.g., "art gallery"), MATTER (e.g., "ceramics tile"), MANNER (e.g., "bus service"). These relations are assumed to hold between the head and the modifiers of the complex expressions extracted previously. The issue is, given a set of relations to be considered, assign the corresponding one, if appropriate, to the components of each complex expression. To do so, the authors assign a feature vector to each complex expression, formed by the hyperonyms of each of its (already disambiguated) components in WordNet. Then, a corpus which has been previously annotated with the type of relation is used to supervisedly learn rules over the vectors that apply to the modifier and the head, who characterise an specific relation. For instance, for the relation MATTER, the learner found the rule that this relation can be assigned when in the vector of the modifier (which includes the modifier's ancestors) the concept "building material" is found (e.g., in the complex expressions *stave church*, or *cobblestone street*).

Learning relationships from search data The use of search results from the web and other search data is becoming and increasingly used resource for learning relationships. Search engines like Google or Ask already provide structured information in their search results when the

user's query is a factual question[17], together with hyperlinks pointing to the pages that contain the supportive evidence for the extracted relation.

The most common way of acquiring information from the web is by means of weakly supervised systems that start with a set of seed words that are related to each other, and extract from the web phrases or dependency chains that connect them in natural language sentences. Next, the context are generalised and applied to new texts. Some example systems that work on surface patters are those described by [Agichtein and Gravano, 2000, Ravichandran and Hovy, 2002, Mann and Yarowsky, 2003, Pantel and Pennacchiotti, 2006, Pennacchiotti and Pantel, 2006], and some that use dependency parses are those described in [Bunescu and Mooney, 2005, Sekine, 2006, Suchanek et al., 2007]. Davidov and Rappoport [2008b], Pasca and Van Durme [2007], Paşca [2007] differ from the previous approaches in the source data, which instead of search results are user query logs.

3.3.2.4 Methods based on annotated corpora

Given the availability of corpora with annotations about relations to extract, from the ACE competitions, it is possible to train machine learning procedures on these datasets. One of the most active lines of research at this moment is the study of different kernel functions to train Support Vector Machines. Zelenko et al. [2003] use a recursive kernel based on parse trees to detect *person-affiliation* and *organisation-place* relationships. Culotta and Sorensen [2004] generalise the previous kernel to dependency trees. Bunescu and Mooney [2005] use a kernel function based on the shortest path in the dependency tree, and Zhao and Grishman [2005] train support vector machines and k-Nearest Neighbours classifiers using a combination of five kernel functions based on syntactic characteristics of the training examples. Huang et al. [2008] study the complementary effects between tree-based kernels and flat kernels.

Related views to the problem are those presented by Kambhatla [2004], using a Maximum Entropy model with attributes from the syntactic analyses, and Jiang and Zhai [2007], which rather than exploring kernel functions focus on identifying the best set of features for the task.

3.3.3 Discussion

The most extended approach to the automatic detection of relationships in text is the use of patterns. These can be simple lexical sequences, enhanced with part of the speech [DeBoni and Manandhar, 2002, Berland and Charniak, 1999], or more complex rules making use of syntax [Harabagiu et al., 1999, Finkelstein-Landau and Morin, 1999].

Amongst the drawbacks of using patterns, the most referred one is the fact that most of the methods depart from manually defined patterns [Alfonseca and Manandhar, 2002b, Kietz et al., 2000]. When defining patterns in a manual way, the rules extracted from observation can vary

[17] An example query is [which is the capital of Nigeria],
http://www.google.com/search?hl=en&q=which+is+the+capital+of+nigeria&btnG=Search
with the result showing as of December 2008.

largely depending on the quantity of data used to elicit a generalisation, the subjective understanding of the context made by the researcher at each case, etc. Some of the works include a metric based on the frequency under which the pattern matched to decide whether the candidate relation discovered between two words can be accepted or not. This type of techniques give way to the data sparseness problem: there is a need to get a significant number of matchings to decide that a relation holds between two concepts, which not always happens, specially when the pattern is too restrictive. In that case, the pattern might not find concepts that should be considered for the relation. In these cases the system has a low recall, indicating that the patterns should be more general. On the other hand, too general patterns return many erroneous matchings, reducing the accuracy of the system. Pattern-based systems usually need an accurate manual tuning to find a compromise between accuracy and recall. The method in [Finkelstein-Landau and Morin, 1999] is an example of how good performance these systems can have when the pattern is rich enough, but, as stated above, this work required such a complete definition of the features characterising the Merge relation that it can be considered an ad-hoc solution, difficult to be ported to other type of relationships. An automatic way to extract, generalise and tune patterns seems to be a desirable solution, as this would guarantee that all the available data is evaluated and broadly used for the generalisation.

Regarding the pure statistical methods, they rely on the frequent occurrence of some features like syntactic relations, part of the speech, etc, holding between a pair of words that indicate that a semantic relation may happen between them. As reported by [Richardson et al., 1998], these systems sometimes need a high amount of manual intervention to validate and tag the extracted relationships. It is desirable that the relationships extraction tool contemplates also the tagging of the relations, as in [Navigli and Velardi, 2003], and this implies using some kind of rules to characterise the type of relations returned by the statistical method.

Finally, dictionaries or encyclopedic texts, or domain-specific texts, seem to be of better use for the extraction of relations than general free-text. First of all, they include salient information, thus avoiding part of the erroneous results that can be extracted from free text. On the other hand, academic dictionary developers use stylistic conventions that result in repetitive formulae in the definitions (for introducing a concept, for expressing its use, location, etc) that in some cases become actual patterns to locate relationships [Novischi, 2002]. Nevertheless, some attention has to be paid to the level of generality of the genus word and other repetitive formulae in definitions: whilst the definition *a dog is an animal that...* would assign a hyponymy relation between *dog* and *animal*, the definition *a mammal is an animal that...* would link *mammal* to *animal*, thus placing *dog* and *mammal* at a same level of generality in the hierarchy. Other drawbacks of dictionaries are the inconsistencies in the usage of word senses in the definitions mentioned in the WSD discussion, which also affect the concepts for which the relationships are extracted, and the circular reference amongst definitions [Rigau, 1998].

The techniques reviewed in this section cope with the problem of extracting semantic relationships, but do not deal with the ambiguity of the words participating in the relation. The task of assigning the extracted relationships to the proper word senses require the use of WSD techniques

as those presented in the previous section.

3.4 Annotating semantic information in Wikipedia

As mentioned above, dictionary and encyclopedia entries usually follow structure and style guidelines that makes it easier to extract information from them than for completely unrestricted corpora. For that reason these data sources have been used in applications of knowledge elicitation for at least two decades. In this context, Wikipedia constitutes a free wide-coverage encyclopedia that can be used as knowledge source for automatically acquiring structured information, using a combination of Natural Language Processing techniques. Furthermore, the availability of structured information in Wikipedia (e.g. category names and a hierarchy of categories, templates, or tabular data) allows systems to complement Natural Language Processing techniques with shallow wrappers.

Concerning the semantic annotations of Wikipedia, there are currently two parallel tendencies: given the proven disposition of volunteers to contribute and extend the contents of the encyclopedia, Völkel et al. [2006] proposed a simple annotation scheme with which non-tech-savvy people can contribute annotations to the existing documents, such as annotating hyperlinks connecting two entries with the predicate that specifies the relationship between them. This proposal, the Semantic Wikipedia, was realized as a MediaWiki extension[18], and there already exist several wikis built on top of it.

On the other hand, other researchers have focused on annotating the contents with automatic techniques. The two approaches are not necessarily exclusive, as the infrastructure built for the Semantic Wikipedia can be used to store the annotations produced automatically, and the annotations marked manually by Semantic Wikipedia users could be used as seeds or training data in order to learn others from the non-annotated entries [Medelyan et al., 2008a].

Some of the best known systems that generate structured content from Wikipedia are YAGO [Suchanek et al., 2007] and DBpedia [Auer et al., 2007, Auer and Lehman, 2007], which focus on exploiting the structured parts of Wikipedia to extract relations. DBpedia analyses Wikipedia's Infoboxes to transform their data into relations in the form of RDF triples, and YAGO maps Wikipedia categories to the WordNet taxonomy. These are large-scale projects that suceed to extract hundreds of thousands of effective relations between concepts pairs and handle a large amount of relations types, e.g. DBpedia includes around 8000 different types of relations.

Concerning the annotation of Wikipedia's unstructured text using Information Extraction techniques, it is popular to try learnt patterns on Wikipedia corpora, sometimes taking advantage of the fact that, for example, the first sentence in an entry usually is a brief definition containing a genus term that can be identified with a morphosyntactic pattern.

One example is the work by Suchanek et al. [2006], who parse all the sentences with a dependency parser, and identify the chains of dependencies connecting two concepts from the training data, and these are used as features to train a Machine Learning model able to classify two can-

[18] http://semantic-mediawiki.org

didate entries as related or unrelated given their dependency-chain features. The relationships studied are birth-date, synonym and instanceOf, and precision and recall values vary much depending on the corpora used, with precision ranging from 33% for the instanceOf relationship on general Wikipedia corpora to 80% for the synonymy relationship on Wikipedia geographical content, and recall varying 33% for the instaceOf relationship on general Wikipedia corpora to 70% for the relationship of birth data on a subcorpus of composers.

With a similar inspiration, Herbelot and Copestake [2006] apply a full parser to Wikipedia articles, and look for copular verbs like *to be*, in order to extract hyponymy relationships. This is just one very conservative rule, similar to one of Hearst's patterns (but using a parse tree instead of a regular-expression pattern), with which they are able to learn relationships with 92% precision at the expense of a very low recall (14%).

Nguyen et al. [2007a,b] use a similar approach, but obtaining the generalised syntactic parses for the relationships of interest automatically. To that aim, they parse the Wikipedia articles, and identify, using a set of heuristics, mentions of the main entity defined. After applying a Named Entity Recogniser module, they identify keywords that are likely to express the relationship and extract sentences containing the positive examples and the keywords. A parse tree generalisation follows to obtain simplified parse trees for each relationship considered. The obtained F-score was 38% (29% precision).

Still following a pattern-based approach, the main contribution of Wang et al. [2007] was to acquire automatically selectional constraints to apply to the placeholders in the patterns for each of the relationships under consideration. Thus, they are able to apply patterns like `X's Y` for relationship like *has-composed*, by forcing the first placeholder to be an artist's name, and the second placeholder to be a music title.

Blohm and Cimiano [2007] use very simple lexical patterns, without part-of-speech or syntactic information, to learn general relations. They depart from a list of seed tuples which hold a relation, and apply different experiments based on iterative induction to learn the patterns from the World Wide Web, from Wikipedia or a combination of both. Their method is evaluated on a Wikipedia subset, yielding a precision of 55% in their best experiment with different 7 relationships.

Also relevant is the work by Zaragoza et al. [2007], in which they have annotated a snapshot of the Wikipedia with syntactic information (using a dependency parser) and named entities.

3.5 Proposed approach and comparison to related work

The particular techniques developed in this work rely on WordNet, Wikipedia and the World Wide Web as knowledge bases to disambiguate encyclopedic entries and to extract the taxonomic *is-a* relation for their classification, as well as several other non-taxonomic relations.The target is to extract semantic information (named entities and relations) to enrich ontologies (in this case WordNet) and annotate web content (in this case Wikipedia).

The approach does not make any particular assumption that would prevent it a priori from

being applied to any type of text. However, these experiments focus on the on-line encyclopedia Wikipedia, because of the following reasons:

- Wikipedia is an encyclopedia, and as such it is organised in entries, each one having a more or less fixed structure: each entry contains a title of the term described, the first paragraph provides a brief definition of that term, and the remaining text further elaborates it. Different modules implemented take advantage of this organisation, as will be seen later.
- It already contains some annotations explicitly provided by the users, such as a (rather noisy) hierarchy of categories, or information on polysemous terms with so-called *disambiguation pages*, which might be useful for extending the proposed architecture.
- Wikipedia has already proven useful as a linguistic resource [Bunescu and Pasca, 2006].

On the other hand, Wikipedia is work in progress, and as such it contains errors, for instance duplicated entries, or dangling links, i.e. hyperlinks to Wikipedia entries that have not been created yet. Throughout this chapter we shall make the distinction between existing Wikipedia pages (those that have been created and contain text) and non-existing Wikipedia pages (those that are pointed to by at least one hyperlink but still have not been created).

Amongst the related works that attempt to generate structured content from Wikipedia, we mentioned YAGO [Suchanek et al., 2007] and DBpedia [Auer et al., 2007, Auer and Lehman, 2007]. Both are different from this work in the fact that they focus on extracting relationships from the structured parts of Wikipedia, like its categories trees or Infoboxes, suceeding in extracting several thousands of relations. Methods (like ours) based on unstructured text usually focus on fewer relations, but have the advantage of being easily portable to other text sets beyond Wikipedia.

Regarding extracting information from the natural language text in Wikipedia, some of the work described in this thesis has been mentioned among the earliest approaches [Medelyan et al., 2008a]. Other related work (cf. Section 3.4) use patterns to extract relations, including e.g. deeper syntactic parsing [Herbelot and Copestake, 2006, Suchanek et al., 2006], anaphora resolution [Nguyen et al., 2007b,a] and selectional constraints [Wang et al., 2007]. Compared to these, our approach is simpler (using part-of-speech and named entities tagging, and only shallow parsing) and thence easier to port across languages. We also use a larger data set for learning and evaluating patterns. Reversely, Blohm and Cimiano [2007] use a lower linguistic processing approach compared to ours, based on lexical patterns without part-of-speech or named entity tagging, but with a substantially larger amount of training data.

Many of the works in disambiguation reviewed in Section 3.1 are not particularly centered, like this work, in the enrichment of ontologies. The method here developed for disambiguation could be classified as a hybrid on methods that use the knowledge source structure and frequency-based methods, one of the most promising lines as discussed at the end of the WSD review chapter. We are focusing on the enrichment of WordNet, a semantic network that already contains a large amount of well structured concepts. Although other works are also based on WordNet ([Agirre and Lopez de Lacalle, 2007, Hawker, 2007, Cai et al., 2007]), we are centering our experiments

in an encyclopedia, which provides a more relevant context to the terms under study than, for instance, free text on the web. The combination of an encyclopedia with WordNet allows the use of a simpler WSD method than those used in the cited works. We make a modification of the Lesk's frequency-based algorithm ([Lesk, 1986]), improved to exploit WodNet's structure by considering hypernyms of the candidate senses during the disambiguation.

As for the extraction of relations, we take two different approaches, one based on the existing WordNet taxonomy to extract taxonomic relations, which allows the classification of concepts in the ontology, and one based on a rote extractor built from free text in the web to extract non taxonomic relations that are completely new in WordNet. Both approaches follow pattern learning techniques, the most extended approach as discussed in Section 3.3.3. This approach has the advantage of avoiding syntactic parsing, sometimes required to obtain good results in pure statistical methods [Maedche and Staab, 2000] and avoids the need to name clusters or statistically-derived relations [Navigli and Velardi, 2003, Caraballo, 1999]. The goal is to keep a low complexity on text processing and to avoid excessive post-processing such as a manual selection of acceptable patterns.

In the case of the taxonomic relations, the *is-a* and *part-of* relations already in WordNet are used as a departing point for a *learning-by-example* approach. Kietz et al. [2000] use also an approach based on WordNet (they use the German version GermaNet) applied to a corporate intranet instead of Wikipedia. Their work uses an approach based on Hearst's method for manual feeding of pattern, which they enrich with heuristics. Contrarily, the method in our work is focused on learning and generalising patterns automatically, trying to avoid an overload in manual and subjective intervention discussed in Section 3.3.3.

Since WordNet 1.7 does not include many varied non-taxonomic relations, the method here selected to extract general relations is that of learning relations from web searches, in particular is based on [Ravichandran and Hovy, 2002]. Web querying in the search of relations avoids the data sparseness problem present when using limited size corpora, as already discussed in the Section of literature review, but introduces the known problem on the patterns quality and ambiguity present when using free text from the web. To tackle that limitation, our method seeks an improvement over Ravichandran's by a new and strict automatic pruning of patterns.

Chapter 4

A Case in Semantics extraction using Wikipedia and WordNet

The previous chapters have introduced the main technologies and open problems in the Semantic Technologies field. Also the need for the development of automatic and semiautomatic tools for ontology population and semantic annotation has been discussed, as well as the current state of global approaches to fulfil this demand. Natural Language Processing (NLP) has been pointed as a potential tool to apply to the automatic extraction of ontological knowledge, and the levels of language analysis required for such application have been remarked. Specifically, the main NLP techniques to perform Word Sense Disambiguation (WSD), Named Entities Recognition (NER) and Relationships Extraction (RE) have been reviewed.

This chapter is dedicated to the primary goals outlined in Section 1.2. In particular, it deals with the design, implementation, testing and evaluation of methods that semi-automatically extract semantics through NLP-based techniques.

The proposed approach implements techniques on Word Sense Disambiguation, Named Entity recognition and Relations Extraction. It takes advantage of the combination of different knowledge sources: Wikipedia, WordNet and the World Wide Web. The Wikipedia entries are disambiguated using a method based on Lesk (see Section 3.1.4.3), and related to WordNet synsets. A new method for learning and generalising surface patterns learnt from Wikipedia and WordNet allows the extraction of the taxonomic *is-a* relation, for the classification of Wikipedia entries that are not included in WordNet. The patterns include lexical information, part-of-speech and named entity categories. In order to extract general relations between terms, this new method is integrated with a procedure similar to that proposed by Ravichandran and Hovy [2002] (see Section 3.3.2.3), enhanced with a new methodology to prune imprecise patterns.

Section 4.1 describes in more detail the approach followed and the system design. The implementation of the system modules and the experiments are presented in detail in Section 4.2 (linguistic processing), Section 4.3 (NER), Section 4.4 (WSD implementation and testing) and Section 4.5 (RE implementation and testing). Finally, Section 4.6 outlines some applications of

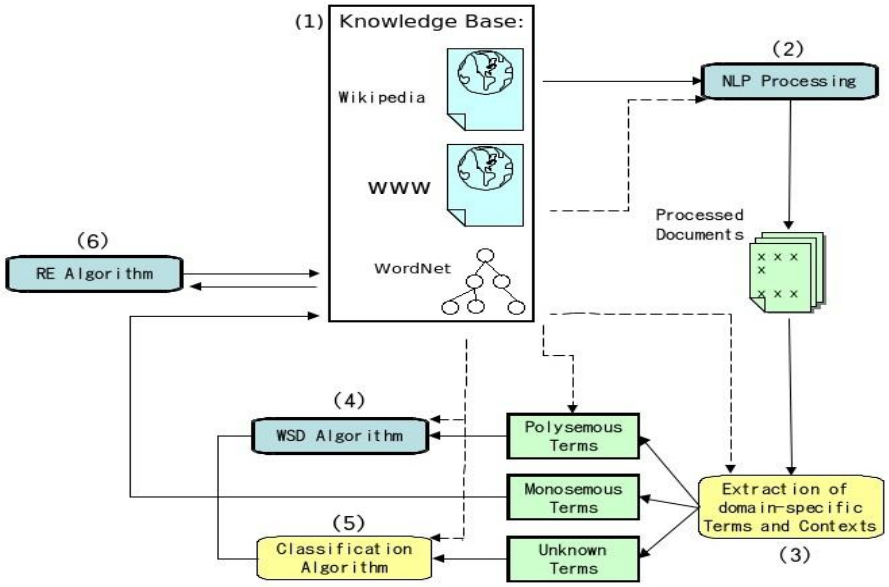

Figure 4.1: Architecture of the system for the automatic annotation of the Wikipedia with semantic relations.

this work.

4.1 System Architecture

The general architecture of the proposed approach for the acquisition and annotation of semantics from Wikipedia is depicted in Figure 4.1. The system proceeds in the following way:

1. A central Knowledge Base stores: (1) As main focus, the Wikipedia entries and all the lexical and semantic annotations obtained about them automatically produced by the system. (2) An ontology, or a lexical semantic network, to which the Wikipedia entries will be associated, in order to have some word knowledge to depart from. The current implementation uses WordNet [Miller, 1995], although other lexical semantic networks would also be valid.(3) Web-based corpora are needed by some of the system modules. These consist in several sets of free-text documents downloaded from the World Wide Web.

2. A static dump of the Wikipedia is processed with NLP tools. All the information obtained in this step is directly encoded in XML inside the downloaded Wikipedia entries, so it can be easily accessed by the following modules.

3. A set of relevant terms is extracted from the Wikipedia pages, including (a) the titles of existing pages; (b) the target of hyperlinks to pages that do not exist yet; and (c) named entities identified by the NLP tools that do not have an associated page, representing names

of people, locations or organisations. These terms are now contrasted with the lexical semantic network stored in the central repository, and classified in three types:

- those terms that do not appear in the semantic network, which are labelled as *unknown* (for the moment),
- those terms that appear both in the Wikipedia and in the semantic network with just one meaning. These are assumed to be monosemous, and to have the same meaning as in the network. For those terms for which there is an entry in the Wikipedia, the entry is immediately associated to the corresponding node in the semantic network,
- those terms that appear with several senses in the semantic network or in Wikipedia, which are labelled as *polysemous* terms.

4. The terms from step 3 that are polysemous are subject to a disambiguation process in order to discover which Wikipedia entries correspond to each of the meanings of those words in WordNet.

5. The terms from step 3 that are labelled as unknown are fed into a classification module that tries to classify them inside WordNet by learning taxonomic relations (hypernymy, *is-a* relations) between them.

6. Furthermore, a procedure to learn non-taxonomic relations is also run.

The following subsections further elaborate on these steps, explaining the technical approach, pseudo-code, experimental samples, evaluation procedures and results for the different modules.

4.2 Linguistic processing

As mentioned, the experiments depart from a static dump of the Wikipedia stored in the Knowledge Base. Prior to the linguistic processing the downloaded entries are pre-processed to get a text format so that those web pages which contain more than one definition are separated into as many files as different senses, the HTML tags corresponding to menus and navigation links are removed, and the page is cleaned from everything except the textual entry.

The computational linguistics tools chosen for the experiments is the wraetlic NLP toolkit [Alfonseca et al., 2006b] that provides information on tokenisation, sentence splitting, stemming, morphological analysis, part-of-speech tagging, Named Entity Recognition and Classification, chunking and partial parsing. The toolkit implements these modules using up to date NLP techiques, being a contribution of this work its extension with a basic Word Sense Disambiguation procedure and one additional procedure for Named Entity Recognition:

- The new WSD procedure is based on the Lesk algorithm [Lesk, 1986], initially developed as baseline for the WSD module in our architecture (see below Section 3.1).

```
<p id="1">
  <s id="2">
    <np appositive="yes" id="1822">
      <np det="definite" head="yes" id="3" number="singular" person="3">
        <w c="w" id="4" pos="DT">The</w>
        <w c="w" id="5" pos="JJ">Asian</w>
        <w c="w" id="6" pos="NN" stem="giant">giant</w>
        <w c="w" head="yes" id="7" pos="NN" stem="hornet">hornet</w>
      </np>
      <np head="yes" id="1820">
        <w c="brackets" id="8" pos="(">(</w>
        <np head="yes" id="9">
          <w c="w" id="10" pos="NNP" stem="Vespa">Vespa</w>
          <w c="w" id="11" pos="NN" stem="mandarinium">mandarinia</w>
        </np>
        <w c="brackets" id="12" pos=")">)</w>
      </np>
    </np>
    <w c="," id="13" pos=",">,</w>
    <np det="definite" id="14" number="singular" person="3">
      <np det="none" id="17" number="singular" person="3" case="genitive">
        <w c="w" id="15" pos="DT">the</w>
        <w c="w" head="yes" id="16" pos="NN" stem="world">world</w>
        <w c="ctr" id="19" pos="POS">'s</w>
      </np>
      <w c="w" id="20" pos="JJS">largest</w>
      <w c="w" head="yes" id="21" pos="NN" stem="hornet">hornet</w>
    </np>
    <w c="," id="22" pos=",">,</w>
    <vbar id="23" tense="finite" time="present">
      <w c="w" head="yes" id="24" lexhead="yes" pos="VBZ" stem="be">is</w>
    </vbar>
    <np det="indefinite" id="25" number="singular" person="3">
      <w c="w" id="26" pos="DT">a</w>
      <w c="w" head="yes" id="27" pos="NN" stem="native">native</w>
    </np>
    <w c="w" id="28" pos="IN">of</w>
    <np det="none" id="29" number="singular" person="3">
      <w c="w" id="30" pos="JJ">temperate</w>
      <w c="w" id="31" pos="CC">and</w>
      <w c="w" head="yes" id="32" pos="JJ">tropical</w>
      <np entity="location" id="33">
        <w c="w" id="34" pos="NNP" stem="Eastern">Eastern</w>
        <w c="w" id="35" pos="NNP" stem="Asium">Asia</w>
      </np>
    </np>
  </s>
</p>
```

Figure 4.2: Sample sentence with shallow syntactic information: *The Asian giant hornet (Vespa mandarinia), the world's largest hornet, is a native of temperate and tropical Eastern Asia.*

- The Name Entity Recognizer in the Wraetlic toolkit included originally a Maximum Entropy classifier and an Error-driven Transformation List Learning system for the identification of the named entities People, Organisations and Locations. These systems can now be combined with an automatically-learnt sure-fire rules system [Alfonseca and Ruiz-Casado, 2005]. The sure-fire rules system was developed as an application of the pattern learning and generalisation algorithm used in the module for Relationships Extraction (see below Section 4.5), which has the advantage of a very high precision for named entities such as People and Location (97 and 96 % respectively) but the drawbacks of a low recall (28 and 19 % respectively) and long processing time.

As indicated, the annotations are encoded in XML inside the Wikipedia entries, which makes it convenient when they are needed for analysis later on. For illustration, Figure 4.2 shows the annotations added to a sample sentence after having been analysed by the tools. As can be seen, the syntactic analysis performed is only partial, as prepositional phrase attachment or coordination are not resolved by the parser.

Using the above-mentioned modules the toolkit was able to process a dump of the English Wikipedia (approx. 1,300,000 articles, at the time when this step was carried out) in about two weeks on a 800 MHz PC laptop with 1 GB of RAM memory.

4.3 Named Entities Recognition

Traditional Term Identification systems [Justeson and Katz, 1995, Cabré et al., 2001] study different properties of domain-dependent texts in order to identify terms that are relevant for a particular domain. Typical approaches combine lexicosyntactic patterns and various statistical analysis of term distributions (whether a term is used uniformly throughout the text or appears in bursts), relative frequency in domain-specific texts compared to general-purpose corpora, and word association metrics to identify multiword expressions.

In this particular case, dealing with the Wikipedia, the main interest was annotating relationships between concepts that are relevant enough to possibly have their own entry in the encyclopedia. Some of these entries may be general-purpose concepts (e.g. *cat* or *house*), others are domain-specific concepts (e.g. pterodactyl or thiocyanic acid), and many others are proper names referring to particular instances of those concepts (e.g. Pterodactyl (film) or William Shakespeare). In order to extract these terms, statistical techniques based on domain-specific corpus-analysis are not very appropriate because in the Wikipedia we can find entries from any possible domain of knowledge. If we compare the relative frequency of these terms with the frequency in a general corpus, such as the British National Corpus, many of the general-purpose terms defined in the Wikipedia will not be extracted as terms, and we want to be able to annotate them with semantic information as well.

Furthermore, word-association metrics like Mutual Information may be helpful to find multiword concepts and instances, but they are not so useful to find terms that appear with a low frequency throughout the encyclopedia, such as the names of not well-known people.

Therefore, the maximum-entropy Named Entity classifier module in the Wraetlic tools was deemed a good approach for this application, and was preferred to the sure-fire module implemented for the same tools because of its better trade-off between precision and recall and its fastest processing time.

In addition, the system takes benefit from the structure of the Wikipedia to find easily most of the terms that are described in the entries:

- The titles of the entries, after some cleanup, are good candidates to be relevant terms.
- All the links to (as yet) non-existing Wikipedia pages, because some user considered that those terms are relevant enough to have their own entries. Also, it might be possible to extract relations between the terms defined in the entries containing those links and the non-existing pages.
- All the Named Entities identified during the linguistic processing step, which are already classified as people, locations and organisations. We assume that any person, place or organisation is potentially important enough as to have its own entry in the Wikipedia.

Therefore, all the terms considered are obtained from either links manually annotated in Wikipedia or the output of the wraetlic tools' Named Entity tagger, which attains an F-score of 96% for people and locations, and 92% for organisations [Alfonseca et al., 2006b]. It is assumed that the list of terms will be sufficiently precise. Even though coverage could be somewhat improved with statistical techniques for identifying domain-dependent terms, the Wikipedia is currently so large that provided a sufficiently big subset of entries the list of entities will be broad enough to perform the semantic extraction task implemented by the subsequent system modules.

Once the named entities are determined, all these terms are compared to WordNet, the lexical semantic network with which the Knowledge Base is initialised, and they are grouped in the three categories mentioned before:

- Those that appear with only one meaning both in Wikipedia (i.e. there is just one entry for them) and in WordNet (i.e. there is only one synonym set). For all of these, we assume that they are monosemous terms and they are used with the same meaning in both places, so the WordNet synset is associated in the Knowledge Base to the Wikipedia entry. The motivation for this is to assume that, if only one sense of the term has been chosen to create a Wikipedia entry and to be included in WordNet, it is probably the most salient meaning of that term, given that both Wikipedia and WordNet are general-purpose resources and are, in principle, not biased towards any particular domain.
- Those that appear with more than one meaning either in the Wikipedia or in WordNet. These are labelled as polysemous, and are later processed by the disambiguation module (Section 4.4).
- Those terms that do not appear in WordNet, which are categorised as unknown: the procedure to classify them in WordNet is described in Section 4.5.

John Smith (Ontario MP)	John Smith (Ontario MPP)	John Smith (UK politician)
John Smith (Welsh politician)	John Smith (US politician)	John Smith (Conservative politician)
John Smith (actor)	John Smith (actor 2)	John Smith (BBC)
John Smith (Clockmaker)	John Smith (comics)	John Smith (filmmaker)
John Smith (guitarist)	John Smith (Scientologist)	John Smith (mathematician)
John Smith (dentist)	John Smith (baseball player)	John Smith (footballer)
John Smith (wrestler)	John Smith (1832-1911)	John Smith (1781-1854)
John Smith (Platonist)	John Smith (missionary)	John Smith (brewer)
John Smith (Medal of Honor, 1880)	John Smith (VC)	

Table 4.1: A (non-thorough) listing of different entries in the Wikipedia about persons called John Smith.

4.4 Word Sense Disambiguation

If there are several entries in the Wikipedia with the same title, or if several WordNet synsets contain the name of an entry, it is necessary to disambiguate the meaning of the title in order to associate the entries and the synsets. Polysemous titles in the Wikipedia can be discovered because either two titles only differ by a brief explanation between parentheses, or because there is a disambiguation page. Table 4.1 shows an example of the first case: there are more than twenty entries about a person called *John Smith*, where the entry titles can be distinguished by the small explanation between parenthesis. In this case, there is also a disambiguation page listing all these entries and a few others, including *John Smith of Jamestown*, the English soldier and colony leader. In this case, there is only one synset in WordNet that contains *John Smith*, so it is necessary to contrast all the mentioned entries with the definition and semantic relations of WordNet's John Smith:

> Smith, John Smith – (English explorer who helped found the colony at Jamestown, Virginia; was said to have been saved by Pocahontas (1580-1631))

The problem of matching Wikipedia entries and WordNet synsets corresponds to a particular case of *Word Sense Disambiguation* (WSD). As introduced in the literature review, similarity metrics between the word to disambiguate and each candidate sense are usually used for carrying out the task: different approaches use co-occurrence information [Manning and Schütze, 2001], all WordNet relations [Hirst and St-Onge, 1998], or just the taxonomic *is-a* relation [Resnik, 1995b], with various success rates. WordNet glosses [Mihalcea and Moldovan, 1999] and Machine Learning algorithms [Grozea, 2004, Lee et al., 2004, Strappavara et al., 2004] are also useful in calculating a semantic similarity.

In this particular application the disambiguation task is easier than a general WSD in unrestricted text. Because both Wikipedia entries and WordNet glosses contain term definitions, those that refer to the same entity will probably highlight the same features, for instance both definitions will contain in many cases the same, or very similar, terms. It is much more difficult to discover whether *John Smith*, used in the middle of a sentence in general text, refers to any of the Wikipedia entries, than to match a particular definition of John Smith with one of the entries'

definitions. That is why the disambiguation accuracy is these experiments was expected to be much higher than the accuracy typically obtained by general-purpose WSD systems.

This case consists in finding a similarity metric between encyclopedia entries and WordNet synsets. If they refer to the same concept, we can expect that there will be much in common between the two definitions. This is the reason why the approach followed is mainly a comparison between the two glosses, inspired in [Lesk, 1986]. It consists of the following steps:

1. Represent the Wikipedia entry as a vector e using the Vector Space Model, where each dimension corresponds to a word, and the coordinate for that dimension is the frequency of the word in the entry.

2. Let $S = \{s_1, s_2, ..., s_n\}$ be the set of WordNet synsets containing the term defined in the Wikipedia entry.

3. Represent each synset s_i as the set of words in its gloss: $G_i = \{t_1, t_2, ..., t_{k_i}\}$, including their frequencies.

4. Let N = 1

5. Extend the sets G_i with the synonym words in each synset s_i and its hyperonyms to a depth of N levels.

6. Weight each term t in every set G_i by comparing the frequency of t in G_i with its frequency in the glosses for the other senses. In this way, a numerical vector v_i, containing the term weights, is calculated for each G_i. In the experiments, two weight functions have been tried: tf·idf and χ^2.

7. Represent each Wikipedia entry as the set E of words in its first paragraph (which is usually a short definition of the term being defined). If the length of the first paragraph is below a threshold θ, continue adding paragraphs until it exceeds it.

8. Apply a greedy algorithm to disambiguate the Wikipedia entries and the WordNet senses: while there are entries that are not disambiguated, choose the pair (w_j, v_i) such that the similarity between w_j and v_i is the largest. Two similarity metrics between the two vectors have been tested: the dot product and the cosine, to check whether the normalisation performed by the cosine could affect the results.

   If there is a tie between two or more senses, N is incremented and the procedure goes back to step 5.

In the final settings, this disambiguation was also extended with a simple procedure to identify the genus word in the definitions, both in WordNet and in the Wikipedia entry [Rigau, 1998], using simple patterns. So, for example, if the Wikipedia entry for *John Smith of Jamestown* would have started with the sentence

> John Smith was an English explorer.

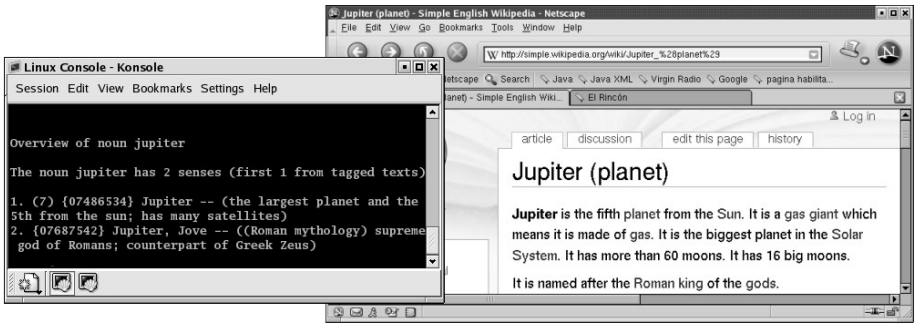

Figure 4.3: Entry for *Jupiter (planet)* in the Wikipedia, and WordNet glosses for the synsets that contain the term *Jupiter*.

then *explorer* would have been identified as the genus word in the entry. Because the WordNet synset containing *John Smith* has as a hyperonym *explorer*, that would have been further positive evidence to associate the entry to the synset. The system currently doubles the similarity score between an encyclopedia entry and a WordNet synset if they have the same genus word or the entry's genus word is a hyperonym of the synset.

4.4.1 WSD Evaluation

As stated above, our disambiguation task is much easier than those usually coped in the SEMEVAL competitions, in which a common test bench of free text is provided for systems comparison. Although some of the best scoring systems in the latest SEMEVAL-2007 were also based on WordNet [Agirre and Lopez de Lacalle, 2007, Hawker, 2007, Cai et al., 2007], applying the information extraction to Wikipedia instead of a freer text is undoubtedly an advantage. For this reason, our results are not fully comparable with the SEMEVAL outputs. Since one of our goals is to enrich WordNet, it seems adequate to measure the improvement in disambiguation precision achieved by our system against the most common sense annotated in the SEMCOR corpus, an information that is part of the WordNet package, as well as reproducing the Lesk method in which ours is founded and applying it to the same test corpus than our method.

The algorithm has been evaluated with a sample of 1841 entries downloaded from the Simple English Wikipedia which senses where manually disambiguated. The version of WordNet used is 1.7. From 1841 Wikipedia terms downloaded, 612 did not appear in WordNet, 631 were found in WordNet with only one possible sense (they are monosemous) and 598 Wikipedia terms were found in WordNet with more than one sense (they are polysemous). Figure 4.3 shows an example of a polysemous term in Wikipedia (Jupiter) that was successfully linked to the correct WordNet sense (planet, ID 07486534). The following evaluations have been performed:

Monosemous terms For these terms, the algorithm just associates each Wikipedia entry with the only WordNet synset containing it. A sample, containing the first 180 monosemous terms from Wikipedia, has been manually evaluated, to check whether this assignment is correct.

Polysemous terms In this case, for each Wikipedia entry there were several candidate senses in WordNet, one of which will be chosen by the algorithm. A sample with the first 180 polysemous terms from Wikipedia was manually annotated with the correct sense. In a few cases, the Wikipedia entry included several senses at the same time, because either (a) the Wikipedia contained two different definitions in the same entry, or (b) the WordNet senses were so fine-grained that they could be considered the same sense. In these cases, all the right senses are annotated, so the algorithm will be considered correct if it chooses one of them (e.g., the Wikipedia entry "Church" mixes two WordNet senses: the church building and the church as group of Christians. In this case, both senses are annotated as correct).

The following baseline experiments and configurations have been tested:

- The first baseline consists of a random assignment.
- The second baseline chooses the most common sense of the word in the sense-tagged SEMCOR corpus. This is a set of texts in which every word has been manually annotated with the sense with which it is used, and it can be used to find which is the most common sense of a word in a general text.
- The third baseline implemented is Lesk's WSD algorithm [Lesk, 1986]. Before applying it, words have been stemmed. Ties between several senses are resolved by choosing SEMCOR's most common sense.
- The three baselines are compared with the procedure implemented for the system WSD module, tested with three possible variations: two choices for the weight function (tf·idf and χ^2), two possible similarity metrics (cosine and dot product), and either stemming or using the lexical form of the words.

The precision for each baseline experiment was computed as the number of correct sense assignments divided by the total number of Wikipedia entries evaluated.

With respect to the monosemous terms, 177 out of the 180 assignments were correct, which means an accuracy of 98.33%. Only in three cases the concept defined by the Wikipedia entry was different to the WordNet sense that contained the same term.

Table 4.2 summarises the accuracy of the different tests for the polysemous terms and for all terms (monosemous and polysemous). These are consistently better than other results reported in WSD, something which, as commented, is attributed to the fact that we are comparing two definitions which are supposed to be similar, rather than comparing a definition with an appearance of a term in a generic text. As can be seen, stemming always improves the results; the best score (83.89%) is statistically significantly higher than any of the scores obtained without stemming at 95% confidence. In many cases, also, tf·idf is better than the χ^2 weight function. Regarding the distance metric, the dot product provides the best result overall, although it does not outperform the cosine in all the configurations. Further experiments using the genus words heuristic improved this result to 87%, using the dot product similarity metric.

	Baselines			*Our approach*							
				Dot product				**Cosine**			
				Stemming		No stemming		Stemming		No stemming	
	Random	SEMCOR	Lesk	tf·idf	χ^2	tf·idf	χ^2	tf·idf	χ^2	tf·idf	χ^2
Polysem.	40.10	65.56	72.78	**83.89**	80.56	77.78	77.78	80.56	81.11	78.33	76.67
All	69.22	81.95	85.56	**91.11**	89.45	88.06	88.06	89.45	89.72	88.33	87.50

Table 4.2: Results obtained for the disambiguation. The first row shows the results only for the polysemous words, and the second one shows the results for all entries in the Wikipedia for which there is at least one synset in WordNet containing the term. The first two columns are the baselines, the third column shows Lesk's algorithm results, and the other eight columns contain the results of the eight configurations tested in our approach.

The accuracies of 98% obtained for monosemous Wikipedia entries and 87% for polysemous entries were the best, therefore the configuration of the WSD module used for the subsequent experiments was based on the dot product, with tf·idf weight and using stemmed and genus word.

4.5 Classification of unknown words and Relations Extraction

The modules for classifying unknown words and for extracting relations share a common technique for learning lexicosyntactic patterns, so they are explained together in this section. The classification of a word inside a taxonomy is approached here as learning taxonomic *is-a* relations between that word and the concepts in the taxonomy, so the problem is understood as a special kind of relation extraction: the extraction of hyponymy, among the many other relations studied.

All the implementation of the relations extraction module is based on the use of lexical patterns. The idea for classifying a new term X in an ontology is to learn patterns such as *"An X is a Y"* or *"X, a kind of Y,"* and try to disambiguate *Y* as some of the concepts in the ontology. If *Y* has a hyperlink to another entry, then it has been already disambiguated in the previous step.

Similarly, for non-taxonomic relations, if we want to learn the birth-date relation, possible patterns are *"X was born in Y at location"* or *"X (location, Y)"*.

Once these patterns have been learnt, they can be applied to the entries categorised as unknown to try to find their hyperonyms. Non-taxonomic relations between the terms can also be obtained with these patterns. The learning process is divided in three steps: *pattern extraction*, *pattern generalisation* and *pattern scoring*, described below.

4.5.1 Pattern extraction

The aim of this step is the extraction of patterns relating two concepts. The process is slightly different for the taxonomic relation hyponymy and for non-taxonomic relations.

In the case of hyponymy relations, the strategy is to find pairs of concepts (t, f) that co-occur in the same sentence, that have already been disambiguated and that have a hypernymy-hyponymy relation in WordNet (Figure 4.4). The process is the following:

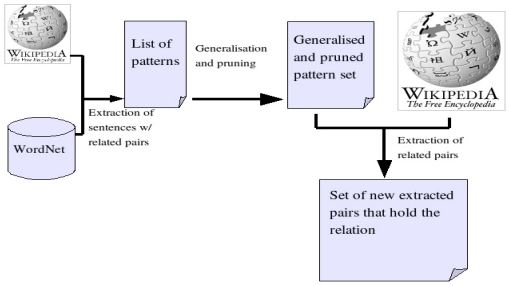

Figure 4.4: Relations extraction for taxonomic and part-of WordNet relations: Wikipedia acts as training corpus. The patterns learnt are applied afterwards to the encyclopedia.

1. For each term t in the Wikipedia, with an entry definition d, every term f is selected, such that
   - t and f co-occur in the same sentence.
   - In d there is a hyperlink pointing to the definition of f.
   - f is a hyperonym of t in WordNet.
2. Extract a context from the sentence, around f and t, including at most five words at their left-hand side, all the words in between them, and at most five words at their right. The context never jumps over sentence boundaries, which are marked with the symbols BOS (*Beginning of sentence*) and EOS (*End of sentence*).
3. The two related terms are marked as `<hook>` and `<target>`.

This way, the existence of terms related as hyponym-hypernym in WordNet serves as seed for the extraction of contexts from Wikipedia. The condition about the hyperlink guarantees that f has already been disambiguated with respect to WordNet with a high precision (see Section 4.4.1).

Additional experiments [Ruiz-Casado et al., 2007] tried to exploit as seeds also other relations included in the WordNet 1.7 version. The algorithm shown above for extracting hyponymy (*is a* relation) was similarly used for its inverse, hypernymy, and for some non taxonomic relations included in Wordnet such as holonymy (*part of* relation) and its inverse meronymy. Antonymy and synonymy are also included in Wordnet, but they were not implemented for this study. The reasons why these relations were discarded are:

- Concerning antonymy, this relation in WordNet does not always refer to the same feature, as sometimes it relates terms that are not true antonyms, like nouns that differ in gender (e.g. *queen* and *king* or in a particular characteristic (e.g. *software* and *hardware*). This would lead to an inconsistent set of patterns to represent the relation, so the use of antonymy from WordNet was discarded.
- With respect to synonymy, it was seen that there are very few sentences in Wikipedia that contain two synonyms together, and, since they are expected to be known by the reader, they

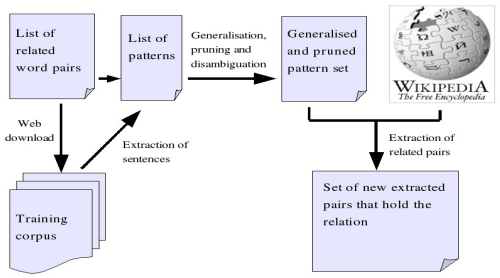

Figure 4.5: General non taxonomic relations extraction: training corpus downloaded from the web, plus seed pairs. The patterns learnt are applied afterwards to the encyclopedia.

are used indistinctly inside the entries. Hence this approach consisting in taking patterns of limited length from Wikipedia was considered not suitable to exploit the synonym relation in WordNet.

A different strategy was needed for the other non-taxonomic relations, since they are not defined in WordNet version 1.7 (Figure 4.5). A set of examples to guide the learning is needed to start the process. For general non-taxonomic relation the procedure selected is also based on limited length patterns, being similar to that of web-based rote extractors [Mann and Yarowsky, 2005, Ravichandran and Hovy, 2002]: for each relation, the user provides the system with a seed list of related pairs. For instance, for the relation birth year, one such pair might be *(Darwin, 1812)*. For each of these pairs, the system:

1. Submits a query to a search engine containing both elements, e.g. *Dickens AND 1812*, and downloads a number of documents to build the training corpus.
2. For any sentence in that corpus containing both elements, the system extracts a context around them in the same way as it was done for the hypernymy relation: at most five words at each side, not crossing sentence boundaries.

The output is, for each relation, a list of free-text patterns that are expected to represent it. In overall, the quality observed for these initial patterns obtained from free web text is naturally lower than those obtained from encyclopedic text.

For illustration, in the case of hypernymy, if the entry for *Dalmatian* contains the sentence *A Dalmatian is a dog that has a white coat with black spots*, the pattern produced would be the following: `A/DT <hook> is/VBZ a/DT <target> that/IN has/VBZ a/DT white/JJ coat/NN.` Note that the words in the pattern are annotated with part-of-speech tags, using the labels defined for the Penn Treebank, and information of Named Entity types is also encoded in the pattern, in the same way than that used for the output of the linguistic processing module (Section 4.2). Figure 4.6 shows examples of patterns that can be found for some relations.

Birth year:
BOS/BOS <hook> (/(<target> -/- number/entity)/) EOS/EOS
BOS/BOS <hook> (/(<target> -/- number/entity)/) British/JJ writer/NN
BOS/BOS <hook> was/VBD born/VBN on/IN the/DT first/JJ of/IN time_expr/entity ,/, <target> ,/, at/IN location/entity ,/, of/IN
BOS/BOS <hook> (/(<target> -/-)/) a/DT web/NN guide/NN
Birth place:
BOS/BOS <hook> was/VBD born/VBN in/IN <target> ,/, in/IN central/JJ location/entity ,/,
BOS/BOS <hook> was/VBD born/VBN in/IN <target> date/entity and/CC moved/VBD to/TO location/entity
BOS/BOS Artist/NN :/, <hook> -/- <target> ,/, location/entity (/(number/entity -/-
BOS/BOS <hook> ,/, born/VBN in/IN <target> on/IN date/entity ,/, worked/VBN as/IN
Author-book:
BOS/BOS <hook> author/NN of/IN <target> EOS/EOS
BOS/BOS Odysseus/NNP :/, Based/VBN on/IN <target> ,/, <hook> 's/POS epic/NN from/IN Greek/JJ mythology/NN
BOS/BOS Background/NN on/IN <target> by/IN <hook> EOS/EOS
did/VBD the/DT circumstances/NNS in/IN which/WDT <hook> wrote/VBD "/" <target> "/" in/IN number/entity ,/, and/CC
Capital-country:
BOS/BOS <hook> is/VBZ the/DT capital/NN of/IN <target> location/entity ,/, location/entity correct/JJ time/NN
BOS/BOS The/DT harbor/NN in/IN <hook> ,/, the/DT capital/NN of/IN <target> ,/, is/VBZ number/entity of/IN location/entity
BOS/BOS <hook> ,/, <target> EOS/EOS
BOS/BOS <hook> ,/, <target> -/- organization/entity EOS/EOS

Figure 4.6: Example patterns extracted from the training corpus for several kinds of relations.

4.5.2 Pattern generalisation

The aim of this step is to identify the portions in common between the patterns, to remove all those terms that they do not have in common and to generalise them.

Pattern generalisation (I): Edit distance calculation In order to measure the likelihood of two patterns a similarity metric is needed.

The procedure used to obtain a similarity metric between two patterns is based on a slightly modified version of the dynamic programming algorithm for *edit-distance* calculation [Wagner and Fischer, 1974] between strings of characters. The *edit distance* between two strings A and B is defined as the minimum number of changes (character insertion, deletion or replacement) that have to be done to the first string in order to obtain the second one. The algorithm can be implemented as filling in a matrix $\mathcal{M}$ with the following procedure:

$$\mathcal{M}[0,0] = 0 \tag{4.1a}$$

$$\mathcal{M}[i,0] = M[\ i-1,0] + 1 \tag{4.1b}$$

$$\mathcal{M}[0,j] = M[0,j-1] + 1 \tag{4.1c}$$

$$\begin{aligned}\mathcal{M}[i,j] = min(&\mathcal{M}[i-1,j-1] + d(A[i],B[j]),\\ &\mathcal{M}[i-1,j] + 1,\\ &\mathcal{M}[i,j-1] + 1)\end{aligned} \tag{4.1d}$$

where $i \epsilon [1...|A|], j \epsilon [1...|B|]$
and

$$d(A[i],B[j]) = \begin{cases} 0 & \text{if } A[i] = B[j] \\ 1 & \text{otherwise} \end{cases}$$

A: `It is a kind of`
B: `It is nice of`

$\mathcal{M}$	**0**	**1**	**2**	**3**	**4**
0	0	1	2	3	4
1	1	0	1	2	3
2	2	1	0	1	2
3	3	2	1	1	2
4	4	3	2	2	2
5	5	4	3	3	2

$\mathcal{D}$	**0**	**1**	**2**	**3**	**4**
0		I	I	I	I
1	R	E	I	I	I
2	R	R	E	I	I
3	R	R	R	U	I
4	R	R	R	R	U
5	R	R	R	R	E

Figure 4.7: Example of the edit distance algorithm. A and B are two word patterns; $\mathcal{M}$ is the matrix in which the edit distance is calculated, and $\mathcal{D}$ is the matrix indicating the choice that produced the minimal distance for each cell in $\mathcal{M}$.

In these equations, M[i,j] will contain the edit distance between the first i elements of A and the first j elements of B. Equation (4.1a) indicates that, if A and B are both empty strings, the edit distance should be 0. Equations (4.1b) and (4.1c) mean that the edit distance between an empty string, and a string with N symbols must be N. Finally, equation (4.1d) uses the fact that, in order to obtain a string[1] $A\sigma$ from a string $B\gamma$, we may proceed in three possible choices:

- We may obtain $A\gamma$ from $B\gamma$, and next substitute γ by σ. If γ and σ are the same, no edition will be required.
- We may obtain $A\sigma\gamma$ from $B\gamma$, and next delete γ at the end.
- We may obtain A from $B\gamma$, and next insert the symbol σ in the end.

In the end, the value at the rightmost lower position of the matrix is the edit distance between both strings. The same algorithm can be implemented for word patterns, if we consider that the basic element of each pattern is not a character but a whole token.

At the same time, while filling matrix $\mathcal{M}$, it is possible to fill in another matrix $\mathcal{D}$, in which it is recorded which of the choices was selected as minimum in equation (4.1d). This can be used afterwards in order to have in mind which were the tokens that both strings had in common, and in which places it was necessary to add, remove or replace tokens. The choices were marked using the following four characters:

- `I` means that it is necessary to insert a token, in order to transform the first string into the second one.
- `R` means that it is necessary to remove a token.
- `E` means that the corresponding tokens are equal, so it is not necessary to edit them.
- `U` means that the corresponding tokens are unequal, so it is necessary to replace one by the other.

[1] $A\sigma$ represents the concatenation of string A with character σ.

Figure 4.7 shows an example for two patterns, A and B, containing respectively 5 and 4 tokens. The first row and the first column in $\mathcal{M}$ would be filled during the initialisation, using Formulae (4.1b) and (4.1c). The corresponding cells in matrix $\mathcal{D}$ are filled in the following way: the first row is all filled with `I`'s, indicating that it is necessary to insert tokens to transform an empty string into B; and the first column is all filled with `R`'s indicating that it is necessary to remove tokens to transform A into an empty string. Next, the remaining cells would be filled by the algorithm, looking, at each step, which is the choice that minimises the edit distance. $\mathcal{M}(5, 4)$ has the value 2, indicating the edit distance between the two complete patterns. For instance, the two editions would be replacing `a` by `nice`, and removing `kind`.

Pattern generalisation (II): Algorithm After calculating the edit distance between two patterns A and B, matrix $\mathcal{D}$ can be used to obtain a generalised pattern, which should maintain the common tokens shared by them. The procedure is the following:

1. Initialise the generalised pattern G as the empty string.
2. Start at the last cell of the matrix $\mathcal{D}(i, j)$. In the example, it would be $\mathcal{D}(5, 4)$.
3. While we have not arrived to $\mathcal{D}(0, 0)$,
   (a) If $(\mathcal{D}(i, j) =$ `E`$)$, then the two patterns contained the same token A[i]=B[j].
      - Set $G = A[i]\ G$
      - Decrement both i and j.
   (b) If $(\mathcal{D}(i, j) =$ `U`$)$, then the two patterns contained a different token.
      - $G = A[i]|B[j]\ G$, where `|` represents a disjunction of both terms.
      - Decrement both i and j.
   (c) If $(\mathcal{D}(i, j) =$ `R`$)$, then the first pattern contained tokens not present in the other.
      - Set $G = *\ G$, where $*$ represents any sequence of terms.
      - Decrement i.
   (d) If $(\mathcal{D}(i, j) =$ `I`$)$, then the second pattern contained tokens not present in the other.
      - Set $G = *\ G$
      - Decrement j

If the algorithm is followed, the patterns in the example will produced the generalised pattern

```
It is a kind   of
It is nice     of
_____________________
It is a|nice * of
```

The wild card $*$ in the pattern can be filled by any combination of words. For example, the pattern above may match phrases such as *It is a kind of*, *It is nice of*, *It is a hyperonym of*, or *It is a type of*. As can be seen, the generalisation of these two rules, as presented above, produces one that can match a wide variety of sentences, but it may be easily mixing together different kinds of relationships between concepts, which is a non desired effect. The challenge was to obtain a degree of generality sufficient to raise a significant number of matchings for a certain relation, but restricting the generality enough to avoid patterns that represent other relations as well as the intended one.

As a first attempt to address over-generalisation, the use of part-of-speech was included in the algorithm.

Pattern generalisation (III): Generalisation with part-of-speech tags The previous example shows that, when two patterns are combined, sometimes the result of the generalisation is far too general, and matches a wide variety of sentences that don't share the same meaning. Therefore, in order to restrict the kinds of patterns that can combine to produce a generalisation, the algorithm has been extended to handle part-of-speech tags. Now, a pattern will be a sequence of terms, and each term will be annotated with a part-of-speech tag, as in the following examples:

(a) `It/PRP is/VBZ a/DT kind/NN of/IN`
(b) `It/PRP is/VBZ nice/JJ of/IN`
(c) `It/PRP is/VBZ the/DT type/NN of/IN`

The edit distance algorithm is modified in the following way: the system only allows replacement actions if the words from the two patterns A and B belong to the same general part-of-speech (nouns, verbs, adjectives, adverbs, etc.). Also, if this is the case, the system considers that there is no edit distance between the two patterns. In this way, two patterns that do not differ in the part-of-speech of any of their words will be considered more similar than other pairs of patterns differing in the part-of-speech of one word. The d function, therefore, is redefined as:

$$d(A[i], B[j]) = \begin{cases} 0 & \text{if } PoS(A[i]) = PoS(B[j]) \\ 1 & \text{otherwise} \end{cases} \tag{4.2}$$

The insertion and deletion actions are defined as before. Therefore, patterns (a) and (b) above would have an edit distance of 2, and the result of their generalisation is:

`It/PRP is/VBZ * of/IN`

On the other hand, the patterns (a) and (c) would have an edit distance of 0, and the result of their generalisation would be the following:

`It/PRP is/VBZ a|the/DT kind|type/NN of/IN`

Pattern generalisation (IV): Generalisation of the whole patterns' set The above describes the generalisation procedure for two patterns. The following algorithm is used in order to generate the whole set of generalised patterns:

1. Store all the patterns in a set $\mathcal{P}$.
2. Initialise a set $\mathcal{R}$ as an empty set.
3. While $\mathcal{P}$ is not empty,
   (a) For each possible pair of patterns, calculate the edit distance between them (allowing for three edit operations: insertion, deletion or replacement).
   (b) Take the two patterns with the smallest distance, p_i and p_j.
   (c) Remove them from $\mathcal{P}$, and add them to $\mathcal{R}$.
   (d) Obtain the generalisation of both, p_g, as described.
   (e) If p_g does not have a wildcard adjacent to the hook or the target, add it to $\mathcal{P}$.
4. Return $\mathcal{R}$

At the end, $\mathcal{R}$ contains all the initial patterns and those obtained while generalising the previous ones. The motivation for step (e) is that, if a pattern contains a wildcard adjacent to either the hook or the target, it will be impossible to know where the hook or the target starts or ends. For instance, when applying the pattern `<hook> is a * <target>` to a text, the wildcard prevents the system from guessing where the hyperonym starts.

As an example, the following are generalisations of hypernymy patterns:

```
    <hook> is/VBZ a/DT <target>

<hook> is/VBZ a/DT type/NN of/IN <target>

<hook> is/VBZ a|the/DT <target> for|in|of|that/IN

A|An|The/DT <hook>/NNP is/VBZ a|the/DT <target> that/WDT

disjunction-of-verbs/VBZ

<hook> is/VBZ the/DT disjunction-of-superlative-adjectives/JJS <target> on/IN Earth|earth/NNP
```

The algorithm to process the whole patterns' set is afterwards slightly modified by scoring and pruning rules, explained next, depending on whether the relation is learnt from WordNet or from the World Wide Web.

4.5.3 Pattern scoring and pruning

As has been seen in the examples, the patterns obtained in the previous step cover all the space between very specific and constrained patterns and very general patterns. Very general patterns may extract many results with a low precision, while very restricted patterns may have a high precision but a small recall. In the scoring step, an estimate of the precision of the patterns is carried out, and the generalised patterns are pruned to those for which the best results are expected.

The differences in the quality of the knowledge source used to generate the initial patterns sets different exigences in how to control the generalised patterns accuracy:

- Hyponymy and holonymy seed patterns were derived from WordNet's related pairs and sentences extracted from Wikipedia (see Figure 4.4). Since in WordNet these relations have been manually annotated and the patterns are extracted from encyclopedic text, a good accuracy was expected for the generalised patterns, and the pruning could be tackled through a simple threshold.

- In the case of general relations the quality of the departing information is lower: the departing corpus of patterns was created by a massive querying of free text on the web (see Figure 4.5), where different senses for very general patterns may easily be mixed. In order to discard too general patterns a more sophisticated approach based on *rote extractors* was selected.

4.5.3.1 Pruning for the WordNet-derived relations

The pruning is done by adding a threshold parameter to the generalisation procedure, in such a way that two patterns with an edit distance exceeding the threshold are not generalised:

1. Store all the patterns in a set $\mathcal{P}$.
2. Initialise a set $\mathcal{R}$ as an empty set.
3. While $\mathcal{P}$ is not empty,
   (a) For each possible pair of patterns, calculate the edit distance between them (allowing for three edit operations: insertion, deletion or replacement).
   (b) Take the two patterns with the smallest distance, p_i and p_j.
   (c) If the edit distance between then exceeds a threshold θ, stop.
   (d) Otherwise,
      i. Remove them from $\mathcal{P}$, and add them to $\mathcal{R}$.
      ii. Obtain the generalisation of both, p_g, as described.
      iii. If p_g does not have a wildcard adjacent to the hook or the target, add it to $\mathcal{P}$.
4. Return $\mathcal{R}$

The purpose of the parameter θ is the following: if no limit is set to the algorithm, ultimately all the rules can be generalised to a single generalisation containing little but an asterisk, which would match almost any text. Thus, it is desirable to stop merging rules when the outcome of the merge is too general and would be source of a large quantity of errors. The value of θ was set empirically to 3. For higher values of θ, the system tries to generalise very different rules, resulting in patterns with many asterisks and few lexical terms. The influence of the threshold over the precision is discussed in detail at Section 4.5.4.

4.5.3.2 Scoring and pruning for the general type relations: Automatic Pattern Scoring

In this case, an estimate of the precision of each pattern belonging to the generalised set is automatically calculated. The starting point was related work [Mann and Yarowsky, 2005, Ravichandran and Hovy, 2002] in which patterns obtained in this way are scored using the following

procedure:

For each pair *(hook,target)* in the seed list:

1. Download a separate corpus from the web, called *hook corpus*, using a query that contains just the hook of the relation.
2. Apply the previous patterns to the hook corpus, calculate the precision of each pattern as the number of times it identifies a target related to the hook divided by the total number of times the pattern appears.

To illustrate this process, let us suppose that we want to learn patterns to identify birth years. We may provide the system the seed pair (*Dickens*, *1812*). From the corpus downloaded for training, the system extracts sentences such as

Dickens was born in 1812
Dickens (1812 - 1870) was an English writer
Dickens (1812 - 1870) wrote Oliver Twist

The system then generates the patterns and identifies that the contexts of the last two sentences are very similar, so it also produces a generalisation of those and appends it to the list:

```
<hook> was born in <target>
<hook> ( <target> - 1870 ) was an English writer
<hook> ( <target> - 1870 ) wrote Oliver Twist
<hook> ( <target> - 1870 )
```

The system needs to estimate automatically the precision of the extracted patterns, in order to keep the best ones. So as to measure these precision values, a hook corpus would be downloaded using the hook *Dickens* as the only query word, and the system would look for appearances of the patterns in this corpus. For every occurrence in which the hook of the relation is Dickens, if the target is 1812 it will be deemed correct, and otherwise it will be deemed incorrect (e.g. in *Dickens was born in Portsmouth*).

The rote extractor approach following the above was tried initially. These early experiments showed that this procedure for calculating the precision of the patterns is unreliable in some cases. For example, the following patterns are reported by Ravichandran and Hovy [2002] for identifying the relations Inventor, Discoverer and Location:

Relation	Prec.	Pattern
Inventor	1.0	<target> 's <hook> and
Inventor	1.0	that <target> 's <hook>
Discoverer	0.91	of <target> 's <hook>
Location	1.0	<target> 's <hook>

In the particular application in which they are used (relation extraction for Question Answering), they are useful because there is initially a question to be answered that indicates whether we are looking for an invention, a discovery or a location. However, if we want to apply them to unrestricted relation extraction, we have the problem that the same pattern, the genitive construction, represents all these relations, apart from the most common use indicating possession.

Relation name	Seed-list	Cardinality	Hook-type	Target-type	Web queries
birth year	birth-date.txt	n:1	entity	entity	$1 was born in $2
death year	death-date.txt	n:1	entity	entity	$1 died in $2
birth place	birth-place.txt	n:1	entity	entity	$1 was born in $2
country-capital	country-capital.txt	1:1	entity	entity	$2 is the capital of $1
author-book	author-book.txt	n:n	entity	unrestricted	$1 is the author of $2
director-film	director-film.txt	1:n	entity	unrestricted	$1 directed $2, $2 directed by $1

Table 4.3: Example rows in the input table for the system.

If patterns like these are so ambiguous, then why do they receive so high a precision estimate in Ravichandran and Hovy's approach? One reason is that the patterns are only evaluated for the same hook for which they were extracted. To illustrate this with an example, let us suppose that we obtain a pattern for the relation *located-at* using the pairs *(New York, Chrysler Building)*. The genitive construction can be extracted from the context *New York's Chrysler Building*. Afterwards, when estimating the precision of this pattern, only sentences containing *<target>'s Chrysler Building* are taken into account. Because of this, most of the pairs extracted by this pattern may extract the target *New York*, apart from a few that extract the name of the architect that built it, *van Allen*. Thus we can expect that the genitive pattern will receive a high precision estimate as a *located-at* pattern.

For the purpose in this work, however, it is desired to collect patterns for several relations such as *writer-book*, *painter-picture*, *director-film*, *actor-film*, and it is needed to make sure that the obtained patterns are **only** applicable to the desired relation. Patterns like *<target> 's <hook>* are very likely to be applicable to all of these relations at the same time, so the rote extractor approach was modified to be able to discard them automatically by assigning the too general patterns a low precision.

Following this purpose, these three improvements to the basic rote extractor were implemented:

1. Collecting not only a *hook corpus* but also a *target corpus* should help in calculating the precision. In the example of the *Chrysler building*, we have seen that in most cases that we look for the pattern *<target>'s Chrysler building* the previous words are *New York*, and so the pattern is considered accurate. However, if we look for the pattern *New York's*, we shall surely find it followed by many different terms representing different relations, and the precision estimate will decrease.
2. Testing the patterns obtained for one relation using the hook and target corpora collected for other relations. For instance, if the genitive construction has been extracted as a possible pattern for the *writer-book* relation, and we apply it to a corpus about painters, the rote extractor can detect that it also extracts pairs with painters and paintings, so that particular pattern will not be very precise for that relation.
3. Many of the pairs extracted by the patterns in the hook corpora were not evaluated at all when the hook in the extracted pair was not present in the seed lists. To overcome this, the web can be queried to check whether the extracted pair might be correct, as shown below.

Automatic Pattern Scoring: implementation In this implementation, the rote extractor starts with a table containing some information about the relations for which we want to learn patterns. This procedure needs a little more information than just the seed list, which is provided as a table in the format displayed in Table 4.3. The data provided for each relation is the following: (a) The **name of the relation**, used for naming the output files containing the patterns; (b) the name of the file containing the **seed list**; (c) the cardinality of the relation. For instance, given that many people can be born on the same year, but for every person there is just one birth year, the cardinality of the relation *birth year* is n:1. On the other hand, several authors may write several books each, so the cardinality would be n:n in this case; (d) the **restrictions** on the linguistic annotations for the hook and the target. These can be of the following three categories: *unrestricted*, if the pattern can extract any sequence of words as hook or target of the relation, *Entity*, if the pattern can extract as hook or target only things of the same entity type as the words in the seed list (as annotated by the Named Entity Recogniser module), or *PoS*, if the pattern can extract as hook or target any sequence of words whose sequence of part-of-speech labels was seen in the training corpus; and (e) a sequence of **queries** that could be used to check, using the web, whether an extracted pair is correct or not.

At this point, the system has already used the seed list to extract and generalise a set of patterns for each of the relations using training corpora as described in Section 4.5.1 and Section 4.5.2. In this case, however, the original patterns before the generalisation step are kept in the pattern set together with their generalisations. Then, the procedure for calculating the patterns' precision is as follows:

1. For every relation,
   (a) For every *hook*, collect a *hook corpus* from the web.
   (b) For every *target*, collect a *target corpus* from the web.
2. For every relation r,
   (a) For every pattern P, collected during training, apply it to every hook and target corpora to extract a set of pairs.
   For every pair $p = (p_h, p_t)$,
   - If it appears in the seed list of r, consider it correct.
   - If it appears in the seed list of other relations, consider it incorrect.
   - If the hook p_h appears in the seed list of r with a different target, and the cardinality is 1:1 or n:1, consider it incorrect.
   - If the target p_t appears in r's seed list with a different hook, and the cardinality is 1:1 or 1:n, consider it incorrect.
   - Otherwise, the seed list does not provide enough information to evaluate p, so a test on the web if performed: for every query provided for r, the system replaces \$1 with p_h and \$2 with p_t, and sends the query to Google. The pair is deemed correct if and only if there is at least one answer.

   The precision of P is estimated as the number of extracted pairs that are assigned as correct divided by the total number of pairs extracted.

In this step, every pattern that does not apply a minimum number of times can also be discarded, to filter out too specific patterns that, while not helping recall, do add computational cost.

Example After collecting and generalising patterns for the relation *director-film*, each pattern is applied to the hook and target corpora collected for every relation. Let us suppose that we want to estimate the precision of the pattern

<target> 's <hook>

and we apply it to the hook and the target corpora for this relation and for *author-book*. Possible pairs extracted are *(Woody Allen, Bananas), (Woody Allen, Without Fears), (Charles Dickens, A Christmas Carol)*. Only the first one is correct. The rote extractor proceeds as follows:

- The first pair appears in the seed list, so it is considered correct.
- Although *Woody Allen* appears as hook in the seed list and *Without Fears* does not appear as target, the second pair is still not considered incorrect because the *directed-by* relation has n:n cardinality.
- The third pair appears in the seed list for *writer-book*, so it is directly marked as incorrect.
- Finally, because the system has not yet made a decision about the second pair, it queries Google with the sequences

  Woody Allen directed Without Fears
  Without Fears directed by Woody Allen

  Because neither of those queries provide any answer, it is considered incorrect.

In this way, it can be expected that the patterns that are equally applicable to several relations, such as *writer-book*, *director-film* or *painter-picture* will attain a low precision because they will extract many incorrect relations from the corpora corresponding to the other relations.

Automatic Pattern Scoring: Evaluation of the patterns and filtering This section studies the results for the step on automatic scoring of non-taxonomic/non-WordNet patterns obtained from the web using the rote extractor approach. Additionally, a manual evaluation of the precision of such patterns is carried out and compared to (a) the precision automatically estimated and (b) the precision calculated following the traditional hook corpus approach from the departing literature.[2]

Table 4.4 quantifies the information handled by the pattern scoring module, at different steps of the procedure. Nineteen relations were considered, and the table shows first the number of seed pairs that were provided as input for each relation, the next column shows the number of extracted patterns from the *hook* and *target* corpora, next the number of generalised patterns and finally the number of patterns after filtering. Note that the generalisation procedure produces new (generalised) patterns that are added to the set of original patterns, but no pattern is removed except when the wildcard is adjacent to the hook or the target. Thus, the set of patterns tends to increase after the generalisation. The filtering criterion was to keep the patterns that applied at least twice on the test corpus.

[2]Note that these results and evaluation correspond to the intermediate step on automatic pattern scoring, while Section 4.5.4 refers to the results and evaluation for the whole system applied to extract relations to Wikipedia.

Relation	Seeds	Extr.	Gener.	Filt.
Birth year	244	2374	4748	30
Death year	216	2178	4356	24
Birth place	169	764	1528	58
Death place	76	295	590	10
Author-book	198	8297	16594	513
Actor-film	49	739	1478	6
Director-film	85	6933	13866	319
Painter-painting	92	597	1194	26
Employee-organisation	62	1667	3334	7
Chief of state	55	1989	3978	15
Soccer player-team	194	4259	8518	84
Soccer team-city	185	180	360	0
Soccer team-manager	43	994	1988	9
Country/region-capital	222	4533	9066	107
Country/region-area	226	762	1524	2
Country/region-population	288	318	636	3
Country/region-borderings	157	6828	13656	479
Country/region-inhabitant	228	2711	5422	58
Country/region-continent	197	1606	3212	52

Table 4.4: Number of seed pairs for each relation, and number of unique patterns in each step.

Concerning the computational complexity, for each pair $(hook, target)$ we need a corpus for each hook and a corpus for each target. For example in the case of birth year, that means 488 corpora in total. In these experiments a maximum of 500 documents per corpus have been downloaded, so the total number of documents is less or equal than 244,000, on which the 4,748 patterns obtained after the generalisation step are evaluated. This procedure required an average of 48 hours for each relationship on a Pentium IV laptop computer (1 GHz, 2 GB RAM).

It is interesting to see that for most relations the reduction of the pruning is very drastic. This is because of two reasons. Firstly, most patterns are far too specific, as they include up to 5 words at each side of the hook and the target, and all the words in between. Only those patterns that have generalised very much, substituting large portions with wildcards or disjunctions are likely to apply to the sentences in the hook and target corpora. Secondly, the samples of the hook and target corpora used are too small for some of the relations to apply, so few patterns apply more than twice.

Concerning the precision estimates, a full evaluation is provided for the *birth-year* relation. Table 4.5 shows in detail the thirty patterns obtained. It can also be seen that some of the patterns with good precision contain the wildcard *. For instance, the first pattern indicates that the presence of any of the words *biography*, *poetry*, etc. anywhere in a sentence before a person name and a date or number between parenthesis is a strong indication that the target is a birth year, regardless the words in between represented by the wildcard.

The third column in the table indicates the number of times that each rule applied in the hook and target corpora, and the next columns indicate the precision of the rule in each of the following

No.	Pattern	Applied	Prec1	Prec2	Real
1	Biography|Hymns|Infography|Life|Love|POETRY|Poetry| Quotations|Search|Sketch|Woolf|charts|genius|kindness| poets/NN */* OF|Of|about|by|for|from|like|of/IN <hook> (/(<target> -/-	6	1.00	1.00	1.00
2	"/'' <hook> (/(<target> -/-	4	1.00	1.00	1.00
3	[BOS]/[BOS] <hook> was/VBD born/VBN about|around|in/IN <target> B.C.|B.C.E|BC/NNP at|in/IN	3	1.00	1.00	1.00
4	[BOS]/[BOS] <hook> was/VBD born/VBN about|around|in/IN <target> B.C.|B.C.E|BC/NNP at|in/IN location/entity	3	1.00	1.00	1.00
5	[BOS]/[BOS] <hook> was/VBD born/VBN around/IN <target> B.C.E/NNP at/IN location/entity ,/, a/DT	3	1.00	1.00	1.00
6	[BOS]/[BOS] <hook> was/VBD born/VBN around|in/IN <target> B.C.|B.C.E/NNP at|in/IN location/entity ,/,	3	1.00	1.00	1.00
7	[BOS]/[BOS] */* ATTRIBUTION|Artist|Author|Authors| Composer|Details|Email|Extractions|Myth|PAL|Person| Quotes|Title|Topic/NNP :/, <hook> (/(<target> -/-	3	1.00	1.00	1.00
8	classical/JJ playwrights/NNS of/IN organisation/entity ,/, <hook> was/VBD born/VBN near/IN location/entity in/IN <target> BCE/NNP ,/, in/IN the/DT village/NN	3	1.00	1.00	1.00
9	[BOS]/[BOS] <hook> (/(<target> -/-)/)	2	1.00	1.00	1.00
10	[BOS]/[BOS] <hook> (/(<target> -|--/-)/)	2	1.00	1.00	1.00
11	[BOS]/[BOS] <hook> (/(<target> person/entity BC/NNP ;/, Greek/NNP :/,	2	1.00	1.00	1.00
12	ACCESS|AND|Alice|Author|Authors|BY|Biography|CARL| Dame|Don|ELIZABETH|(...)|web|writer|writerMuriel| years/NNP <hook> (/(<target> -|- -/-	8	0.75	1.00	
13	-/- <hook> (/(<target> -/-	3	0.67	1.00	0.67
14	-|--/- <hook> (/(<target> -/-	3	0.67	1.00	0.67
15	[BOS]/[BOS] <hook> (/(<target> -/-	60	0.62	1.00	0.81
16	[BOS]/[BOS] <hook> (/(<target> -/- */*)/)	60	0.62	1.00	0.81
17	[BOS]/[BOS] <hook> (/(<target> -|--/-	60	0.62	1.00	0.81
18	,|:/, <hook> (/(<target> -/-	32	0.41	0.67	0.28
19	[BOS]/[BOS] <hook> ,/, */* (/(<target> -|--/-	15	0.40	1.00	0.67
20	,|:|;/, <hook> (/(<target> -|--/-	34	0.38	0.67	0.29
21	AND|Alice|Authors|Biography|Dame|Don|ELIZABETH|Email| Fiction|Frances|GEORGE|Home|I.|Introduction|Jean|L|Neben| PAL|PAULA|Percy|Playwrights|Poets|Sir|Stanisaw|Stanislaw| W.|WILLIAM|feedback|history|writer/NNP <hook> (/(<target> -/-	3	0.33	n/a	0.67
22	AND|Frances|Percy|Sir/NNP <hook> (/(<target> -/-	3	0.33	n/a	0.67
23	Alice|Authors|Biography|Dame|Don|ELIZABETH|Email| Fiction|Frances|GEORGE|Home|I.|Introduction|Jean| L|Neben|PAL|PAULA|Percy|Playwrights|Poets|Sir|Stanisaw| Stanislaw|W.|WILLIAM|feedback|history|writer/NN <hook> (/(<target> -/-	3	0.33	n/a	0.67
24	[BOS]/[BOS] <hook> ,|:/, */* ,|:/, <target> -/-	7	0.28	0.67	0.43
25	[BOS]/[BOS] <hook> ,|:/, <target> -/-	36	0.19	1.00	0.11
26	[BOS]/[BOS] <hook> ,/, */* (/(<target>)/)	20	0.15	0.33	0.10
27	[BOS]/[BOS] <target> <hook> ,/,	18	0.00	n/a	0.00
28	In|On|on/IN <target> ,/, <hook> grew|was/VBD	17	0.00	0.00	0.00
29	In|On|on/IN <target> ,/, <hook> grew|was|went/VBD	17	0.00	0.00	0.00
30	[BOS]/[BOS] <hook> ,/, */* DE|SARAH|VON|dramatist|novelist| playwright|poet/NNP (/(<target> -/-	3	0.00	n/a	1.0
	TOTAL	436	0.46	0.84	0.54

Table 4.5: Patterns for the relation *birth year*, results extracted by each, precision estimated with this automatic procedure (Prec1) and with the traditional hook corpus approach (Prec2), and precision evaluated by hand (Real).

cases:

- As estimated by the modified precision calculation presented in this work, complete with target and hook corpora, cardinality and web queries (Prec1).
- As estimated by the traditional hook corpus approach (Prec2). Here, cardinality is not taken into account, patterns are evaluated only on the hook corpora from the same relation, and those pairs whose hook is not in the seed list are ignored.
- The real precision of the rule (real). In order to obtain this metric, two different annotators evaluated a sample of 200 pairs independently, and the precision was estimated from the pairs in which they agreed.

As can be seen, in most of the cases the new modified procedure here proposed produces lower precision estimates. If we calculate the total precision of all the rules altogether, shown in the last row of the table, we can see that, with the traditional hook corpus approach, the whole set of rules would be considered to have a total precision of 0.84, while that estimate decreases

Relation	Prec1	Prec2	Real
Birth year	0.46	0.84	0.54
Death year	0.29	0.55	0.38
Birth place	0.65	0.36	0.84
Death place	0.82	1.00	0.96
Author-book	0.07	0.26	0.03
Actor-film	0.07	1.00	0.02
Director-film	0.03	0.26	0.01
Painter-painting	0.05	0.35	0.17
Employee-organisation	0.31	1.00	0.33
Chief of state	0.11	-	0.00
Soccer player-team	0.07	1.00	0.08
Soccer team-city	-	-	-
Soccer team-manager	0.61	1.00	0.83
Country/region-capital city	0.12	0.23	0.12
Country/region-area	0.09	1.00	0.06
Country/region-population	1.00	1.00	1.00
Country/region-borderings	0.17	1.00	0.15
Country-inhabitant	0.01	0.80	0.01
Country-continent	0.16	0.07	0.00

Table 4.6: Precision estimates for the whole set of extracted pairs by *all* the filtered rules and all the relations.

sharply to 0.46 with the new modified precision estimate. This value is nearer the actual precision of 0.54 evaluated by hand. Note that the precision estimated by the new procedure is even lower than the real precision of the patterns due to the fact that the web queries consider unknown pairs as incorrect unless they appear on the web exactly in the format of the query in the input table. Specially for not very well-known people, we cannot expect that all of them will appear on the web following the pattern *"X was born in date"*, so the web estimates tend to be over-conservative.

Table 4.6 shows the precision estimates for every pair extracted with all the rules, using both procedures (the new and the traditional), for the non-taxonomic relations. The real precision has been estimated by sampling randomly 200 pairs for each relation and evaluating them by hand, as explained above for the *birth year* relation. There was an overall 96.29% agreement (Kappa[3]=0.926).As can be observed, the precision estimate of the whole set of rules for each relation (*Prec 1*) is nearer to the real precision (*Real*) in many more cases than using the traditional hook corpus approach (*Prec 2*). Note as well that the precisions indicated in the table refer to the complete set of data here used, including all the pairs extracted by all the rules, some of which are very precise, but some of which are very imprecise. For end applications, after the scoring, it is possible to select only those rules whose individual precision estimate is above a defined value, which expectedly will boost the overall results in precision.

[3]Kappa=(Observed agreement-chance agreement)/(Number pairs-chance agreement)

4.5.4 Identification of new relations in Wikipedia

Once we have a set of patterns for each semantic relationship, the extraction procedure is simple. Each pattern will contain:

- A flag indicating whether the hook appears before the target, or vice-versa.
- A left-hand part, consisting of the words at the left of the hook or the target, whichever appears first.
- A middle part, with the words between the hook and the target.
- A right-hand part.

Given a set of patterns for a particular relation, the procedure to obtain new related pairs is as follows:

1. Download the corpus that should be annotated from the web.
2. Clean the HTML code, remove menus and images
3. Process the textual corpus with NLP tools:
   (a) Tokenise the text and split sentences.
   (b) Apply a part-of-speech tagger.
   (c) Stem nouns and verbs.
   (d) Identify Named Entities.
   (e) Chunk Noun Phrases.
4. For every pattern,
   (a) For each sentence in the corpus,
      i. Look for the left-hand-side context of the pattern in the sentence.
      ii. Look for the middle context.
      iii. Look for the right-hand-side context.
      iv. Extract the words in between for each context section, and check that either the sequence of PoS tags or the entity type are correct. If so, output the relationship.

4.5.5 Experimental settings and Evaluation

The methods and system implementation described in the previous sections were tested for all the relations introduced so far. This section presents and evaluates the results.

Most of the techniques on relations extraction date from the last decade, being on more focus since 2005, and, to my knowledge, there are not well-established test benches for comparing systems. SEMEVAL-2007 in its Task 4 presented a test bench for classification of relations,

but it was focused to nominals excluding named entities, hence it is not applicable to this work. Therefore, a subset of the extracted relations have been manually evaluated. In the case of the non-taxonomic relations, since our pattern extraction uses a rote extractor and is grounded on Ravichandarn and Hovy's previous work, our approach was compared with theirs by reproducing the original method and applying both to the same test data.

A summary of our results is as follows:

- In the case of the four WordNet-drawn relations, the number of extracted relations and their precision depend on the threshold parameter that stops the generalisation process (see Section 4.5.3.1). With a parameter equal to 3 (maximum distance of 3 insertions, deletions or replacements between patterns) the total number of extracted related pairs, excluding those that were already existing in WordNet is 2622 with a mean precision for the 4 relations of 63,3%.

- In the case of the non taxonomic relations learnt through the rote extractor procedure, an experiment pruning patterns below the automatic score of 60% produced a total of 16064 related pairs, with a mean precision for all the relations of 84,4%.

These are average values and it must be noted that neither all the patterns nor all the relations attained the same precision. Detailed figures on corpus sizes, number of relations extracted and resulting precision are given in the following subsections.

4.5.5.1 Extraction of related pairs from Wikipedia using the WordNet relations

For hyponymy, hypernymy, holonymy and meronymy, the algorithm has been experimented and evaluated in a corpus of 6090 entries downloaded from the simple English Wikipedia, roughly 35 MB of textual data after cleaning HTML.

The initial set of patterns before the generalisation resulted in 485 sentences for hyponymy, 213 for hypernymy, 562 for holonymy and 509 for meronymy. When analysing these patterns, however, it was found that both for hypernymy and meronymy most of the sentences extracted only contained the *hook*, with no contextual information around it, while the *target* was not captured. The reason was unveiled by examining the web pages:

- In the case of hyponyms and holonyms, it is very common to express the relationship with natural language, with expressions such as *A dog is a mammal*, or *A wheel is part of a car*. These sentences are easily analysed by the linguistic tools and successfully captured into the patterns.

- On the other hand, when describing hyperonyms and meronyms, their hyponyms and holonyms were found to be usually expressed with enumerations, which tend to be formatted as HTML bullet lists, e.g.:

  A dispensing hydraulic system is composed of:

  - *Storage tank*

Threshold	No. of patterns	Known Relations	New relations	Prec.
1	19	681	1284	72.43%
3	26	951	2162	65.12%
5	23	700	2095	53.13%
7	26	721	2158	52.78%
9	30	729	2217	51.87%

Table 4.7: Results obtained when extracting hyponymy relationships using different thresholds to stop the generalisation of the rules.

- *Pump*
- *Distribution lines*
- *etc.*

In these cases the sentence splitter chunks each hyponym and each holonym as belonging to a separate sentence, thus failing to capture them together in a pattern.

The corpus to extract the new relations was the same used to learn them. The number of generalised patterns and extracted pairs depends on the distance threshold used to stop the generalisation process, as explained in Section 4.5.3.1. All the new extracted pairs were manually evaluated by two judges with inter-judge agreement of 95%. The precision is calculated as number of correct pairs divided by total number of new[4] pairs. A pair was deemed correct when it actually holds the semantic relation under study.

Hyponymy Table 4.7 shows the results obtained for several values of the threshold θ that governs when to stop generalising the patterns. The number of patterns refers to generalised (merged) patterns. With threshold 1, only patterns that have an edit distance less or equal to 1 can be merged. The system output consisted of 19 generalised patterns in this case. Note that all the patterns that had not merged with any other are discarded for the result of the generalisation. The 19 patterns extracted a total of 1965 relationships, out of which 681 were already present in WordNet, and the remaining 1284 were evaluated by hand, with an overall precision of 72.43%.

As the threshold increases, more rules can be merged, because their edit distance becomes lower than the threshold, so we in general a larger set of generalised rules is obtained. Also, because more rules have been generalised, the number of results increases with threshold 3, and remains rather stable for higher thresholds. On the other hand, as can be expected, the precision drops as we generalise the rules more and more, because we obtain rules with fewer content words that can apply in other contexts not related to hyponymy.

Table 4.8 shows some of the rules extracted with the threshold 3. The pattern that applied most often is the classical hyponymy copular expression, `hook is a target`, which relates a concept with its hyperonym (rules 7, 8 and 10). There are several versions of this pattern, allowing for extra tokens before and in between, and providing a long list of adjectives that may

[4]Pairs that were already included in WordNet do not compute in the precision calculation. Also repetitions (pairs extracted more than once by different rules) are excluded

No.	Match	Prec.	Rule
1	6	1.0	HOOK/NN is/VBZ a/DT type/NN of/IN TARGET
2	1	1.0	HOOK/NNP is/VBZ the/DT */* common|largest/JJS TARGET on/IN Earth|earth/NNP
3	1	1.0	The/DT HOOK/NNP are|is/VBZ */* big/JJ TARGET in/IN eastern/JJ North/NNP America/NNP
4	1	1.0	HOOK|Isotopes|Jupiter|Neptune|Saturn|Uranus|Venus/NNS are|is/VBP */* different|eighth|fifth|first|second|seventh|sixth|small/JJ TARGET from|in|of/IN the/DT */* Ocean|Sun|element|sun|year/NN
5	152	0.92	HOOK/NNP is|was/VBD a|an/DT British|English|alcoholic|non-metal|old/JJ TARGET
6	6	0.83	The/DT HOOK/NNP is/VBZ a|the/DT TARGET around|for|in|of/IN the/DT */* Party|Pole|States|Yorkshire|tree/NNP
7	574	0.79	HOOK/NN is/VBZ a/DT TARGET
8	579	0.74	*/*, HOOK/NN is/VBZ a|an/DT TARGET
9	29	0.66	HOOK/NN is/VBZ a/DT */* branch|drink|piece|sheet|type/NN of/IN TARGET
10	639	0.49	HOOK/NNP is/VBZ a|the/DT TARGET for|in|of|that/IN */*
11	7	0.43	HOOK/NN came|is/VBZ */* a|an/DT TARGET drink|family/NN
12	36	0.42	TARGET of/IN the/DT Year/NN
13	35	0.17	Earth/NNP 's/POS TARGET
14	78	0.17	HOOK/NN is|use/VBP coins|part/NNS as|of/IN TARGET
15	18	0.0	HOOK TARGET List/NN of/IN colors/NNS
(9 more rules)			
25	0	n/a	The/DT language/NN called/VBD HOOK/NNP is/VBZ one/CD of/IN the/DT language|languages/NNS that/WDT came/VBD from/IN the/DT TARGET language/NN
26	0	n/a	A|An|The/DT HOOK/NNP is/VBZ a|the/DT TARGET that/WDT connects|has|helps|lets/VBZ */* computers|letter|plants|run/NNS */*

Table 4.8: Some of the rules obtained for the relation of hyponymy (threshold 3). Columns indicate the number of the rules, the new results produced by each rule, its precision and the text of the rule.

appear in the definition.

Secondly, there are also patterns which have been extracted because of the writing style of a particular contributor in Wikipedia. For instance, there are several entries about months in the years, and all of them contain a variant of the sentence *XXX is the n-th month in the year*. A similar pattern is used to describe planets in the Milky Way (*XXX is the n-th planet from the Sun*). Rule 4 shows a pattern generalised from those sentences. Other example is that of colours, all of which contain the same sentence, *List of colors*, in their definition (rule 15). In the training corpus, every entry containing that sentence was a hyponym of the concept *color*.

Finally, rules 25 and 26 have been displayed as examples of too specific rules that, because they can only match in very particular contexts, have not been able to identify any hyponymy relationship apart from those that were already in WordNet.

Threshold	No. of patterns	Known Relations	New relations	Precision
1	1	1	0	n/a
3	4	1	0	n/a
5	5	2	16	50%
7	9	9	28	32.14%
9	10	15	77	27.27%

Table 4.9: Results obtained when extracting hypernymy relationships using different thresholds to stop the generalisation of the rules.

Amongst the most common mistakes produced by these rules the following should be noted:

- Errors due to the choice of a modifying PP rather than taking the NP to which it modifies. For example, from the sentence *the man with the telescope is the leader*, the word *telescope* would be chosen as hyponym of *leader*. To correct these errors, the patterns should also include syntactic information.

- Invalid information obtained from erroneous sentences, such as *the U.K. is a communist republic*. The Wikipedia is a supervised Encyclopedia, but the erroneous information introduced by the authors may persist for a few days before it is noticed and removed.

- Typographic errors, e.g. *Swedish* is classified as a hyponym of *launge* from the text:

  > Swedish is a person or a object that comes from the country Sweden. It's like English and England. It can also be the *launge* that is spoken in Sweden

Some of the extracted pairs contain terms that are not existing in WordNet, like *Rochdale F. C.*, classified as a *club*, *Ijtihad* as a *war*, *Bambuco* as a *music*, and *Llanfairpwllgwyngyllgogerychwyrndrobwllllantysiliogogogoch* as a *village*. Also, new hyponymy relations can be found for words existing in WordNet, for instance *Paris* and *Athens*, as the capital towns in France and Greece, appear in WordNet as hyponyms of *capital* and now have a new hyponymy relationship to *city*.

Hypernymy Concerning hypernymy, as commented before, it is usually expressed in the Wikipedia with enumerations, which are not handled properly by the pattern-matching procedure. Consequently, there were very few patterns to use, and those available were very specific. Table 4.9 shows the results of the evaluation for five threshold values. As can be seen, with thresholds 1 and 3, the obtained patterns can just identify one already-known relationship. Using thresholds 5, 7 and 9, the system produced several new results, but with a low precision. A linguistic sentence splitter with ability to process enumerations should be necessary to overcome this problem.

Holonymy The case of holonymy is similar to that of hyponymy. The results are shown in Table 4.10. As can be seen, as we increase the threshold on the edit distance so that two rules

Threshold	No. of patterns	Known Relations	New relations	Precision
1	19	134	79	70.89%
3	22	207	336	59.82%
5	14	304	1746	50.63%
7	15	307	2979	33.43%
9	21	313	3300	31.67%

Table 4.10: Results obtained when extracting holonymy relationships using different thresholds to stop the generalisation of the rules.

are allowed to be merged, we obtain more general rules that can extract more results, but with a lower precision.

Table 4.11 shows some of the rules for holonymy. Most of the *member part-of* and *substance part-of* relations were rightly extracted by the first few rules in the table, which match sentences such as *X is in Y* or *X is a part of Y*. However, they also extracted some wrong relations.

Interestingly, most of the patterns focused on locations, as we can see in rules 1, 3, 5, 6, 7, 8, 9, 11, 12, 13 and 14. A possible explanation is the large number of entries describing villages, cities and counties in the Wikipedia.

In the case of holonymy, several common errors were identified:

- An important source of errors was the lack of a multi-word expression recogniser. Many of the part-of relations that appear in Wikipedia are relations between instances, and a large portion of them have multi-word names. For instance, the application of the set of patterns to the sentence

  Oahu is the third largest of the Hawaiian Islands

  returns the relation *Oahu is part of Islands*, because *Hawaiian_Islands* has not been previously identified as a multi-word named entity. Other erroneous examples are: (a) *kidney* as part of *system*, and not *urinary system*; and (b) *Jan Peter Balkenende* as part of *party* rather than *Christian CDA party*.

- Other errors were due to orthographic errors in the Wikipedia entry (e.g. *Lourve* instead of *Louvre*) and relations of holonymy which held in the past, but which are not true by now, such as *New York City is part of Holland* or *Caribbean Sea is part of Spain*.

- Finally, some errors are also due to the polysemy of the words in the pattern. For instance, the following pattern,

  ```
  HOOK/NNP is/VBZ a|the/DT capital|city|country|province|state/NN in|of/IN TARGET
  ```

  extracts erroneously, from the following sentences:

  (1) Plasma is a state of matter when the bonds between molecular particles are broken and subatomic particles are all lumped in together.
  (2) Weather is the state of the atmosphere at any given time

No.	Match	Prec.	Rule
1	1	1.0	HOOK/NNP */* the/DT */* capital|city/NN of/IN TARGET ,/, */* Japan|city/NN
2	2	1.0	Some/DT TARGET also/RB have/VBP hair/NN like/IN this/DT ,/, and/CC people/NNS sometimes/RB also/RB call/VB this/DT hair/NN a/DT HOOK/NN
3	32	0.75	HOOK/NNP is/VBZ a/DT city|province/NN in/IN TARGET
4	331	0.73	HOOK/NNP is/VBZ a|an|the/DT */* in|of/IN the/DT TARGET
5	104	0.60	HOOK/NNP is|makes|means|was/VBZ */* a|the/DT */* States|corner|countries|country|layer|part|parts|planet/NNS in|of|that/IN */* a|the/DT TARGET
6	18	0.56	HOOK/NNP Countries|city|country|follower/NNS in|of/IN */* East|Southeast|Southeastern|West|faith|world/NN TARGET
7	851	0.45	HOOK/NNP South|capital|city|continent|country| county|fact|state/NN */* as|in|of/IN TARGET
8	396	0.41	HOOK/NNP Things|city|member|north|part|planets|state/NNS in|of/IN the/DT TARGET
9	5	0.4	HOOK/NNP is|was/VBZ a/DT */* country|part|river/NN in|of/IN */* eastern|north|northern/JJ TARGET
10	5	0.4	It/PRP is/VBZ part/NN of/IN the/DT TARGET
11	1	0.0	It/PRP is/VBZ in/IN central|southwest/JJ TARGET
12	0	n/a	HOOK/NNP is/VBZ a|the/DT capital|country/NN */* between|of/IN TARGET and/CC */* Europe|city/NNP
13	0	n/a	The/DT */* Kingdom|Republic|part/NN */* of/IN HOOK/NNP is/VBZ */* a|the/DT country|middle/NN in|of/IN the/DT continent|middle|southwest/NN of/IN TARGET
14	0	n/a	HOOK/NNP (/(Cornish|German|Icelandic|Welsh/NNP */* Bayern|Caerdydd|Kernow|island/NNP)/) is/VBZ a|the/DT */* city|country|county|part/NN in|of/IN TARGET

Table 4.11: Rules obtained for the relation of holonymy (threshold 5), ordered by precision. Columns indicate the rules' number, number of new results found, precision and pattern.

the relationships between *Plasma* and *matter*, and between *weather* and *atmosphere*. This error stems from the fact that *state* is not used with the sense of territorial division, but with the senses, respectively, of *state of matter* and *the way something is with respect to its main attributes*.

Meronymy Concerning the last relation from WordNet studied, meronymy, even though it is also represented quite often with enumerations in the Wikipedia, the results are rather better than those of hypernymy. The results are shown in Table 4.12. The number of results is lower than the case of hyponymy and holonymy, but the accuracy, for the different threshold values, follows a similar behaviour. The precision is very high with threshold 1 (although the number of new results is very low), and decreases as the threshold increases.

Threshold	No. of patterns	Known Relations	New relations	Precision
1	8	32	10	100%
3	10	74	124	62.90%
5	10	78	147	56.46%
7	14	84	473	40.59%
9	18	95	494	40.89%

Table 4.12: Results obtained when extracting meronymy relationships using different thresholds to stop the generalisation of the rules.

4.5.5.2 Extraction of related pairs from Wikipedia, for non taxonomic relations using the rote extractor approach

The modified rote extractor approach was used to extract non-taxonomic relations from Wikipedia. For this experiment, a dump of the encyclopedia including 20,075 entries was used, totalling roughly 460 MB after cleaning the HTML files. Using the scored patterns for the relations studied in Section 4.5.3.2, the sets were pruned to only those patterns which were automatically scored at 60% or higher. This reduced the number of relations considered to thirteen, as for some relations none of the patterns succeeded to reach that score: Person's birth and death year, birth and death place, author-work, director-film, painter-painting, soccer player-team, country's area and population, country/location-borders, country-inhabitant and country-continent. The patterns for these relations are shown in Table 4.14 through Table 4.26

In order to collect a corpus from Wikipedia with good chances to contain many entries holding those relations, a recursive web download starting from the following Wikipedia entries was performed:

- *Prime Minister*, that contains hyperlinks to Prime Ministers from many countries.
- *Lists of authors*, that contains hyperlinks to several lists of writers according to various organising criteria.
- *Lists of actors*, that contains hyperlinks to several lists of actors.
- *List of football (soccer) players*, containing hyperlinks to many entries about players.
- *List of national capitals*, containing the names of national capitals and countries in the world.

From those initial pages, all the links have been followed up to a depth of 3–4 hyperlinks, having collected this way the total 20,075 encyclopedia entries mentioned.

Table 4.13 summarizes the number of patterns considered for each relation, results extracted (pairs of related terms), and the precision attained. The precision has been estimated by correcting manually at least 100 results from each relationship. The precision was computed as correct pairs by total pairs, excluding repeated pairs.

In general the precision through manual evaluation is well above the 60% minimum automatic score set for the rules for most of the relations. The precision is above 90% for the birth data and

Relation	Nr. Patterns	Extracted pairs	Precision
Birth-year	17	10865	96.0%
Death-year	5	25	88.0%
Birth-place	31	1720	91.0%
Death-place	9	259	73.0%
Author-work	12	526	34.6%
Director-film	3	0	n/a
Painter-painting	1	0	n/a
Soccer player-team	8	3	66.6%
Country/region-area	1	1	100.0%
Country/region-population	3	0	n/a
Country/location-borders	114	2664	44.0%
Country-inhabitant	2	1	0.0%
Country-continent	4	0	n/a

Table 4.13: Number of pruned patterns for each relationship, number of pairs extracted by each pattern set, and precision.

falls down to 35-40% for data that proved more difficult to process. The mean precision is 84.4%, averaged for the total 16064 extracted pairs. As pointed out, the automatic scoring procedure is conservative and tends to underestimate the actual precision of the patterns.

The quantity of extracted pairs is, as expected, correlated with the number of patterns left after the pruning. Some relations are represented by such scarce set of patterns that they return just a few related pairs, if any. However, it must also be noted that some relations naturally appear with higher frequency in these corpora: almost every encyclopedic entry about a person will include his birth and death date and places, along with less frequent properties (e.g. whether the person in an artist, a country president or a football player). Hence the system is capable to extract many more pairs for birth and death data, even when the number of patterns after pruning is similar to that of other less productive relations.

Birth Year and Death Year The precision for birth year and death year is high, 96% and 88% respectively, because they are usually expressed with very fixed patterns, and years and dates are entities that are very easily recognised. The few errors are mainly due to the following two cases:

- Named Entity tagging mistakes, e.g. a TV series mistagged as a person, where the years in which it has been shown are taken as birth and death date.
- Names of persons that held a title (e.g. king or president) during a period of time, that is mistakenly considered as their life span.

As can be seen in Table 4.14 and Table 4.15, the patterns for the birth year relation take advantage on the usual annotation of life span between parentheses. The number of results for death year is lower and they correspond to persons from ancient times, e.g. (Aristotle, 322BC), (Julius Caesar, 44BC). The reason is that only patterns with the "Before Christ" acronym succeeded to reach the score of 0.6 or more.

No.	Match	Prec.	Rule
1	4	1.0	"/'' HOOK (/(TARGET -/-
2	3	0.66	-/- HOOK (/(TARGET -/-
3	3	0.66	-|--/- HOOK (/(TARGET -/-
4	8	0.75	ACCESS|AND|Alice|Author|Authors|BY|Biography|CARL|Dame|Don| ELIZABETH|Email|Fiction|Frances|GEORGE|Home|I.|INDEX| Introduction|Jean|L|Ludovico|Mastery|Modern|NEALE|NOTES|Neben| PAL|PAULA|POETRY|Percy|Philosophy|Playwrights|Poets|Savage| Search|Sir|Sketch|Stanisaw|Stanislaw|W.|WILLIAM|activist|artist| author|critic|dramatist|editor|feedback|found.see|francs| grandfather|history|husband|icon|journalist|novelist|page| patriot|playwright|poet|scholar|sterreichischen|stuff|teacher| thetext|von|web|writer|writerMuriel|years/NNP HOOK (/(TARGET -|--/-
5	6	1.00	Biography|Hymns|Infography|Life|Love|POETRY|Poetry| Quotations|Search|Sketch|Woolf|charts|genius|kindness|poets/NN */* OF| Of|about|by|for|from|like| of/IN HOOK (/(TARGET -/-
6	60	0.62	[BOS]/[BOS] HOOK (/(TARGET -/-
7	2	1.00	[BOS]/[BOS] HOOK (/(TARGET -/-)/)
8	60	0.62	[BOS]/[BOS] HOOK (/(TARGET -/- */*)/)
9	60	0.62	[BOS]/[BOS] HOOK (/(TARGET -|--/-
10	2	1.00	[BOS]/[BOS] HOOK (/(TARGET -|--/-)/)
11	2	1.00	[BOS]/[BOS] HOOK (/(TARGET person/entity BC/NNP ;/, Greek/NNP :/,
12	3	1.00	[BOS]/[BOS] HOOK was/VBD born/VBN about|around|in/IN TARGET B.C.|B.C.E|BC/NNP at|in/IN
(5 more rules)			

Table 4.14: Rules kept for the relation Birth Year after pruning (automatic precision scoring 0.6 or more). Columns indicate the rules' number, number of matches at the training corpus, precision and pattern.

Birth Place and Death Place These two relations attained also a high precision (91% and 73%) and number of extracted pairs. Table 4.16 and Table 4.17 show the set of rules used for the extraction of related pairs. The most common error was mistaking places were a particular person grew or developed an activity during his life by birth or death places. The origin of this failure is that, during the training to acquire the patterns, some of the places given as seed correspond to locations where the person was born but also lived, even in some cases a person could be born and die at the same place. This effect can be seen for instance in rules nr. 10, 11 and 12 in Table 4.16 for birth place, and rules nr. 6 to 9 in Table 4.17, where the words "died" or "grew" are retained in birth place patterns, and "grew" or "spent" are retained for death places. Most of these patterns are ruled out by the rote extractor at the pruning phase, when a pattern that matches two kinds of relations are assigned a low score as explained in Section 4.5.3.2, but still in some cases the system was not capable to detect the duplicity, hence the patterns are scored high and pass the filtering. The combination with other relations during the training (e.g. place of residence)

No.	Match	Prec.	Rule
1	10	0.60	`HOOK (/( */* -/- TARGET B.C|B.C.|B.c|BC|BCE|bc/NNP`
2	10	0.60	`[BOS]/[BOS] HOOK (/( */* -/- TARGET B.C|B.c|BC|BCE|bc/NNP`
3	2	1.00	`according/VBG to/TO traditional/JJ legend/NN ,/, HOOK was/VBD killed/VBN in/IN TARGET BC/NNP when/WRB an/DT */* (/(`
4	2	1.00	`according/VBG to/TO traditional/JJ legend/NN ,/, HOOK was/VBD killed/VBN in/IN TARGET BC/NNP when/WRB an/DT eagle/JJ (/(`
5	2	1.00	`according/VBG to/TO traditional/JJ legend/NN ,/, HOOK was/VBD killed/VBN in/IN TARGET BC/NNP when/WRB an/DT eagle/NN (/(`

Table 4.15: Rules after pruning for the relation Death Year.

and a larger training and pruning corpus would increase the probability to detect the duplicated extraction and avoid this failure mode.

Authoring relations The relations Author-Book, Painter-Painting and Director-Film proved to be more difficult to identify.

For Painting and Film Direction only one and three patterns, respectively, reached the 0.6 score (Table 4.18 and Table 4.19) which was insufficient to extract any related pair in the corpus. Upon examining the patterns left out in the filtering, it was seen that many rules which contain very good hints to find the intended relation had been filtered, for instance:

```
,/, HOOK describes|directed|filmed|pushes|set|shot/VBD TARGET
about|as|in/IN
```

was a good pattern candidate because it retains the words “directed”, “filmed” or “shot”, but received a low score and was filtered out. The scarcity of training data is found again as root of this problem: low scores are assigned due to a lack of matching in the training corpus, which would be corrected by processing a larger amount of data.

An additional difficulty observed was that authors’ entries in the Wikipedia usually include a section summarising all the works (filmography, paintings lists, etc). These lists are expressed as an HTML bullet lists or in a separate frame (*Infobox*). In this way, because the information is already semi-structured, the textual patterns cannot apply. It should be easier to extract that data using other simpler procedures that take benefit of the structure of the entry.

Author-Book succeeded to extract 526 pairs with a precision of 34.6%. Still after the rigorous pruning it could be observed the problem, mentioned in the previous sections, that some patterns are applicable for many kinds of relationships at the same time. For example, in Table 4.20, rules nr. 1 and the genitive variations at rules 8 and 12 may extract related and unrelated pairs indistinctly:

No.	Match	Prec.	Rule
1	5	1.00	,/, HOOK was/VBD born/VBN in/IN TARGET ,/,
2	5	1.00	,/, HOOK was/VBD born/VBN in/IN TARGET ,/, location/entity
3	6	0.83	,|:/, HOOK was/VBD born/VBN */* in/IN TARGET ,/,
4	3	1.00	ALFRED|AUTHOR|Actor|Bing|General|Kirk|LORD|MONTGOMERY|Marlon| Michelangelo|TOM|biography|musician|playwright/NN HOOK is|was/VBD */* at|in/IN TARGET ,/,
5	3	1.00	ALFRED|AUTHOR|Actor|Bing|Marlon|TOM|musician|playwright/NN HOOK is|was/VBD born/VBN */* at|in/IN TARGET ,/,
6	3	1.00	ALFRED|AUTHOR|Actor|Bing|Marlon|musician/NNP HOOK is|was/VBD born/VBN at|in/IN TARGET ,/,
7	3	1.00	AUTHOR|Actor|Bing|Marlon|musician/NN HOOK is|was/VBD born/VBN in/IN TARGET ,/, location/entity
8	3	1.00	AUTHOR|Bing|Marlon|musician/NN HOOK was/VBD born/VBN in/IN TARGET ,/, location/entity
9	2	1.00	[BOS]/[BOS] HOOK ,/, born/VBN */* in/IN TARGET ,/,
10	61	0.74	[BOS]/[BOS] HOOK attended|died|grew|was/VBD */* at|in/IN TARGET ,/,
11	61	0.74	[BOS]/[BOS] HOOK died|grew|was/VBD */* at|in/IN TARGET ,/,
12	61	0.74	[BOS]/[BOS] HOOK died|was/VBD */* at|in/IN TARGET ,/,
(18 more rules)			
31	3	1.00	person/entity winner/NN and/CC country/NN musician/NN HOOK was/VBD born/VBN in/IN TARGET ,/, location/entity ,/, on/IN date/entity

Table 4.16: Rules after pruning for the relation Birth Place.

> [...]by 1741 Garrick was the talk of the theatrical scene for his performance in William Shakespeare 's Richard III.

Rule 12 rightly extracted the pair *(William Shakespeare, Richard III)*.

> [...]has been widely remembered as the high point in Puckett 's career.

Rule 12 wrongly extracted the pair *(Puckett, career)*.

Soccer Player-Team This relation was represented by eight very specific rules (see Table 4.21) that matched only three times in the Wikipedia corpus, which makes the 66.6% precision attained not very reliable.

In order to study further the effect of very general patterns, an additional experiment was carried out for this relation: the full set of patterns from the generalisation step were considered (84 in total), and were applied only to the soccer corpus. This corpus contains the download departing from the Wikipedia entry *List of football players*, 629 entries (approx. 9MB after cleaning the files). The precision of the patterns reached 69.3% when they are applied only to the entries about soccer players, but the figure falls down to near 7% when applied to the

No.	Match	Prec.	Rule
1	10	0.80	[BOS]/[BOS] HOOK died/VBD */* in/IN TARGET ,/,
2	4	1.00	[BOS]/[BOS] HOOK died/VBD in/IN TARGET ,/,
3	4	1.00	[BOS]/[BOS] HOOK died/VBD in/IN TARGET ,/, location/entity
4	6	0.66	[BOS]/[BOS] HOOK died/VBD of|on/IN */* in/IN TARGET ,/,
5	3	0.66	[BOS]/[BOS] HOOK died/VBD on/IN date/entity */* in/IN TARGET ,/, location/entity
6	7	1.00	[BOS]/[BOS] HOOK died|grew|passed/VBD */* at|in/IN TARGET ,/,
7	13	0.85	[BOS]/[BOS] HOOK died|grew|passed/VBD */* in/IN TARGET ,/,
8	7	1.00	[BOS]/[BOS] HOOK died|grew|passed|spent/VBD */* at|in/IN TARGET ,/,
9	13	0.85	[BOS]/[BOS] HOOK died|passed|spent/VBD */* in/IN TARGET ,/,

Table 4.17: Rules after pruning for the relation Death Place.

No.	Match	Prec.	Rule
1	3	1.00	The/DT HOOK by/IN location/entity TARGET [EOS]/[EOS]

Table 4.18: Rules after pruning for the relation Painter-painting.

whole Wikipedia corpus collected. This means that there are patterns that, in the domain of soccer, usually indicate the relationship between the player and its club, but in other contexts they may be conveying a different meaning. One of these patterns is the already mentioned genitive construction. In sports articles, when this construction is found between an organisation and a person is usually expressing the *player-team* relation, as in *Liverpool's Fowler*. But it also extracted many wrong pairs from documents belonging to different topics.

Country/Region relations and properties The relations of countries and regions with names of inhabitants, continent, area and population (see patterns in Table 4.22 to Table 4.26) did not extract a significant amount of pairs, partly because the rules were few and too specific. The main two reasons for the poor number of results are: (a) data such area or population is rather found in a table (*Infoboxes*) than at the textual entry, and (b) the number of patters is too low. A larger training corpus would allow to keep more patterns in the pruned set.

In the case of finding the bordering countries or regions for a location, the set of patterns was large, with 114 rules above the 0.6 score, which resulted in 2664 identified relations. The patterns for Region-Bordering region are listed in Table 4.24.The precision was 44%, being the most frequent sources of failure:

- About half of the wrong pairs were Named Entity tagging mistakes, particularly proper names tagged as locations.
- Too general rules.

No.	Match	Prec.	Rule
1	2	1.00	(/(HOOK 's/POS */* '/'' TARGET '/''
2	2	1.00	[BOS]/[BOS] organisation/entity HOOK 's/POS film|movie/NN '/'' TARGET '/'' fits|is/VBZ
3	2	1.00	[BOS]/[BOS] organisation/entity HOOK 's/POS film|movie/NN */* '/'' TARGET '/'' fits|is|provides/VBZ

Table 4.19: Rules after pruning for the relation Director-Film.

No.	Match	Prec.	Rule
1	4	0.75	)/) HOOK ,|:/, TARGET (/(
2	4	1.0	HOOK completed|finished|published|wrote/VBD TARGET ,/,
3	4	1.00	HOOK published|wrote/VBD TARGET ,/,
4	3	0.66	Aldous|Books|Comedy|D.|Download|Dr|Elizabeth|Emily|Index|J.M.|Jean|Omar|Rider|Sir|Victor/NNP HOOK 's/POS TARGET [EOS]/[EOS]
5	3	0.66	Complete|Critical|More|authentic|existential/JJ TARGET of|on/IN HOOK [EOS]/[EOS]
6	2	1.00	Complete|More|authentic/JJ TARGET of/IN HOOK [EOS]/[EOS]
7	3	0.66	The|the/DT TARGET contain|is|provided/VBZ */* of/IN HOOK 's/POS
8	3	0.66	[BOS]/[BOS] TARGET is/VBZ */* the/DT */* in|of/IN HOOK 's/POS
9	3	0.66	[BOS]/[BOS] HOOK 's/POS TARGET continue|established|included|remained|represents|were/VBD
10	2	1.00	for/IN :/, HOOK ,/, TARGET [EOS]/[EOS]
11	3	1.00	for|of|on/IN organisation/entity HOOK 's/POS TARGET [EOS]/[EOS]
12	5	0.60	in/IN HOOK 's/POS TARGET [EOS]/[EOS]

Table 4.20: Rules after pruning for the relation Author-book

- Regions of geographical proximity but not bordering, like in the sentence *Kazakhstan is divided into 14 provinces and the two municipal districts of Almaty and Astana*, where the two districts are not adjoining.

No.	Match	Prec.	Rule
1	2	1.00	Abou|Gary|Keeper|defender|full-back|goalkeeper|keeper| midfielder/NNP TARGET is|joined|joins/VBZ HOOK at|from|in|on| through/IN
2	2	1.00	Abou|defender|full-back|goalkeeper|keeper|midfielder/NN TARGET joined/VBD HOOK at|from|in/IN
3	4	1.00	FOOTBALL/NN NEWS/NN organisation/entity VIEWS/NNS --/- TARGET -/- HOOK [EOS]/[EOS]
4	2	1.00	[BOS]/[BOS] TARGET adds|is|ran|set|was|won/VBD */* current|delighted|first|former|long|new|next/JJ HOOK injury|loanee|player|record|squad|t|task|trainee/NN
5	2	1.00	after|against|as|by|from|like|on|that|while/IN HOOK 's/POS TARGET are|blocked|fell|found|had|has|have|headed|kept| knocked|leaves|left|met|plays|proved|punched|was|went/VBZ
6	2	1.00	forced/VBD a/DT fine/NN save/VB from/IN HOOK goalkeeper/NN TARGET [EOS]/[EOS]
7	2	1.00	forced|has/VBZ */* from|in/IN HOOK goalkeeper/NN TARGET [EOS]/[EOS]
8	3	0.67	for|from|in/IN HOOK fullback|goalkeeper|midfielder|skipper/NN TARGET [EOS]/[EOS]

Table 4.21: Rules after pruning for the relation Soccer Player-Team.

No.	Match	Prec.	Rule
1	2	1.00	[BOS]/[BOS] HOOK covers|has|is/VBZ a|an/DT */* area|island|total/NN of/IN TARGET km|sq/NNP

Table 4.22: Rules after pruning for the relation Country/Region-Area.

4.6 Applications

The previous sections in this chapter presented the experiments for the detection of entities in Wikipedia, their disambiguation and classification through the hyponymy relation with respect to WordNet, and the further extraction of non taxonomic relations, using a system for Semantics Extraction that combines Wikipedia, WordNet and the World Wide Web as knowledge source. The results and techniques here presented have applicability in different fields of Natural Language Processing and Semantic Technologies. In particular, the following have been proposed in different publications arising from this work:

4.6.0.3 Ontology Enrichment.

The method here presented for disambiguating encyclopedic entries against the lexico-semantic network WordNet provides a link between the WordNet glosses and the encyclopedic knowledge, allowing an extension of the textual data that defines the WordNet concepts [Ruiz-Casado et al., 2005a]. The enrichment of the short dictionary-type WordNet glosses with larger encyclopedic textual entries (see Figure 4.8) represents a data extension which might aid NLP-based tasks like

No.	Match	Prec.	Rule
1	2	1.00	`[BOS]/[BOS] HOOK has/VBZ a/DT population/NN */* of/IN TARGET ,/,`
2	2	1.00	`[BOS]/[BOS] HOOK has/VBZ a/DT population/NN of/IN TARGET ,/, */* area|non-nationals|people/NNS`
3	2	1.00	`[BOS]/[BOS] HOOK has/VBZ a/DT population/NN of/IN TARGET ,/, a/DT total/JJ land/NN area/NN`

Table 4.23: Rules after pruning for the relation Country/Area-Population.

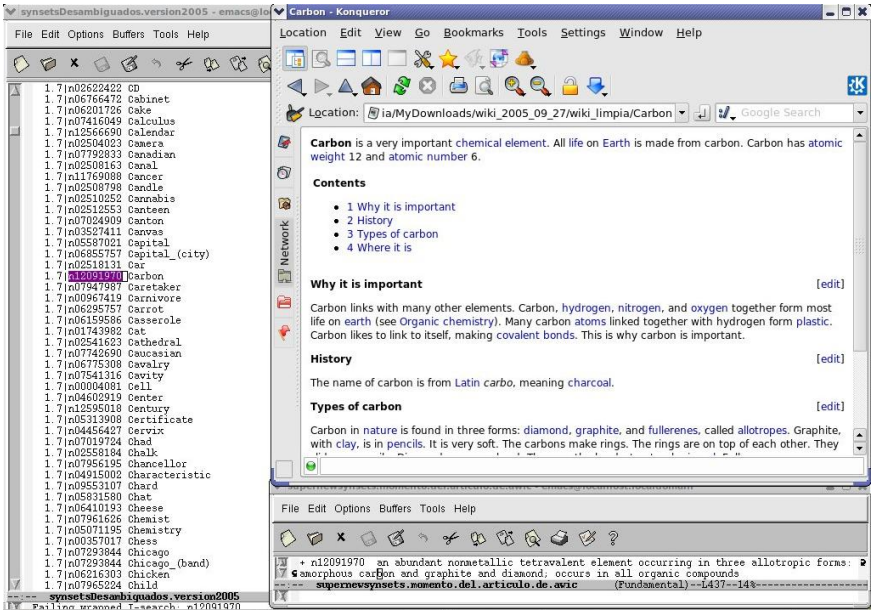

Figure 4.8: WordNet glosses enrichment. Left: WordNet concept codes linked to Wikipedia Entries. Right, up: Text from Wikipedia entry "Carbon". Right, down: WordNet's dictionary definition for "Carbon".

Automatic Text Summarization or Question Answering. These tasks have already been tackled using WordNet glosses [Rus, 2002]. WordNet glosses have been criticised sometimes, as they do not follow any common pattern and some of them are not very informative, with higher extent for the multilingual EuroWordNet where many glosses are nonexistent. Transposing the multilingual Wikipedia to multilingual WordNet using the glosses extension method here proposed might help to solve this problem.

4.6.0.4 Automatically generated list pages.

Wikipedia contains listing pages ordered into *categories*. Among many others, we can find lists of countries, cities, popular people like actors or football players, or lists of events that happened at a given year. A simple experiment to illustrate how to generate these lists automatically was done applying the set of patterns for birth and death year to the Wikipedia corpus used for the non

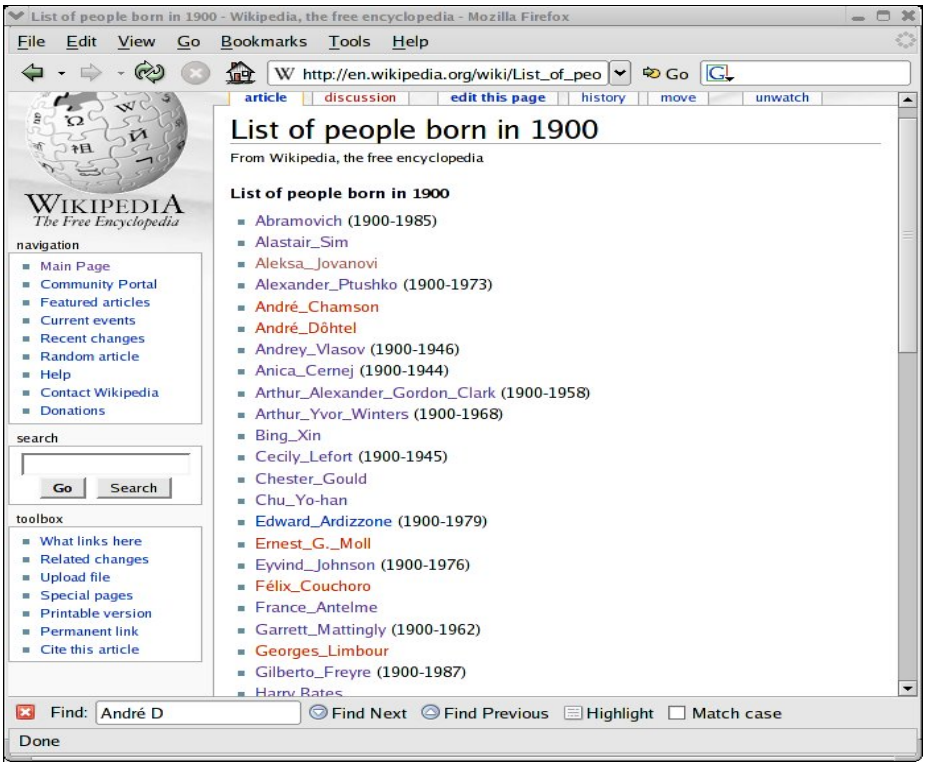

Figure 4.9: A layout for the page generated automatically containing people that were born in 1900, after a few manual corrections.

taxonomic relations extraction [Ruiz-Casado et al., 2006]. One side of the relation is filled with the year in focus: the example chosen is the list of people born the year 1900. A total of 57 people had been annotated with that birth date. The system was able to obtain the list automatically and, in the cases in which the death date was available as well, it was added as additional information. A manual evaluation of the generated list revealed the following:

- Two out of the 57 people were not people, due to errors in the Named Entity recogniser. These were eliminated by hand for the layout example.
- One birth date was erroneous, and it was really the date of publication of a book about that person. That entry was also removed.
- One person had been extracted with two different (but correct) spellings: Julian Green and Julien Green, so they were merged in one.
- From the remaining 53 people, 14 did not have an associated Wikipedia entry. They had been extracted from other lists in which they were mentioned, but their biography is not available yet. These were left inside the list, as it is useful information.
- Finally, three people in the list had ambiguous names, so they were directed to disambiguation pages. It was easy to modify the hyperlink so they pointed directly to the particular

Figure 4.10: Wikipedia entry for Aristotle, as shown in the *Edit* tab of WikiMedia with SemWiki annotations for birth and death year, and birth place.

person with that name that had been born in 1900.

Figure 4.9 shows a possible layout for the generated page. Note that, in the Wikipedia, there are special categories to group all the people born every year. So, the category called *Category:1900_birth* contains all the people for which someone has categorised their entries as having been born in 1900.

The original Wikipedia list contained 640 people born in that year, all of them with an entry in the Wikipedia which has been classified into this category *by hand*. Our procedure extracted the data automatically, collecting 57 people[5]. The generated list identifies four people born in that year that were still not inside the category, and 14 people that were not listed because at the moment nobody had yet written their entries.

In the same way, it should be easy to create or extend list pages using other criteria, such as famous people born in a certain city, or people that died at a certain age.

4.6.0.5 The Semantic Wikipedia.

Chapter 2 pointed out the recent efforts focusing on the creation of a Semantic Wikipedia, that is, a semantically annotated and searchable Wikipedia. Like in other Semantic Web environments, also a Semantic Wikipedia needs to tackle the annotation bottleneck. With more of 2 million articles by the time this text is written, enriching the English Wikipedia with semantic tags might be

[5]Note that the experiment does not process the whole English Wikipedia (more than one million entries at the time this experiment was carried out) but a small subset containing around 20,000 entries.

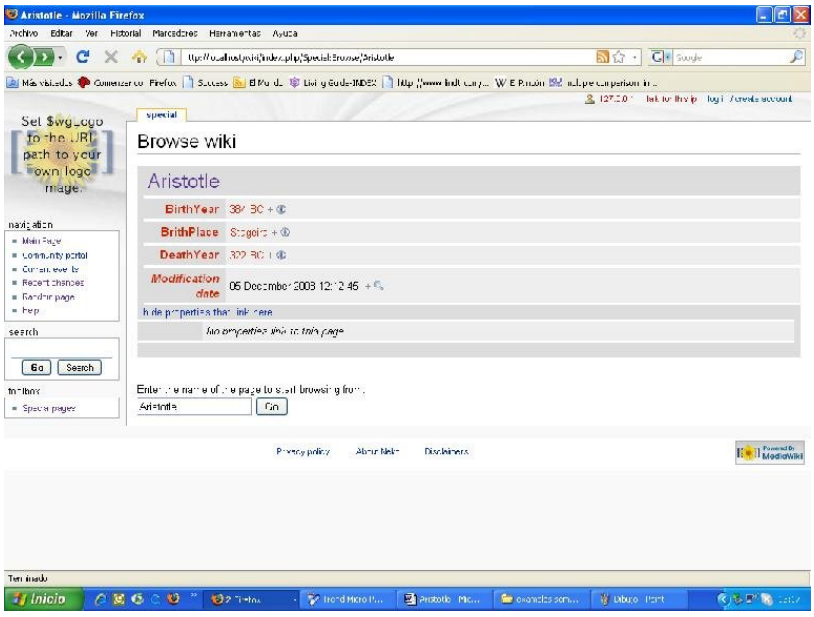

Figure 4.11: Navigation panel for Aristotle's annotated properties and relationships.

aided by the methods and results developed through this work. The opportunity of blending semi-automated semantics extractors like our NLP-based system with semantic wiki environments for annotation were introduced in Ruiz-Casado et al. [2006, 2008a], also mentioned by other authors ([Weld et al., 2008]). There is a mutual benefit when joining these two fields: on the one hand the Wiki philosophy of lowering the technical barriers can successfully be applied to enriching Wiki documents with semantic tags. Manual annotation has the advantage of being accurate, but the drawback of needing a vast amount of work for large databases. On the other hand, NLP-based systems can extract automatically a very large amount of semantic data, but any automatic procedure would produce some amount of mistakes. These mistakes could be corrected by the Wiki contributors, nevertheless, and the automatic tags may well serve as examples to guide the annotators in producing additional tags. An example using the results from the system presented in this work and the Semantic MediaWiki software[6] has been tried. Figure 4.10 illustrates the annotated entry for *Aristotle*. The semantic tags are easily added at the *Edit* section of the entries. In the example of Aristotle, our system automatically extracted the birth and death year, and the birth place. Figure 4.11 illustrates the navigation panel for the semantic properties. The navigation through the semantic tags of a particular Encyclopedia entry is provided by the Semantic MediaWiki extension, as well as the possibility to make (annealed) querying for a(several) defined property.

[6] A free extension of MediaWiki, the wiki systems powering Wikipedia. http://semantic-mediawiki.org

No.	Match	Prec.	Rule
1	4	0.75	,/, TARGET and/CC HOOK agreed|concluded|have|issued|signed|were/VBP
2	4	0.75	,|:/, TARGET and/CC HOOK agreed|are|commit|concluded|decided|have|issued|signed|were/VBP
3	3	1.00	,|:/, TARGET and/CC HOOK agreed|ceased|faced|formalized|had|set|signed/VBD
4	5	0.80	,|:/, TARGET and/CC HOOK agree|agreed|are|began|ceased|commit|concluded|confirm|decided|do|engaged|established|faced|fell|formalized|fought|grew|had|have|held|issued|restored|set|shake|signed|were/VBD
5	4	0.75	,|:/, TARGET and/CC HOOK agree|agreed|are|began|ceased|commit|concluded|decided|faced|formalized|grew|had|have|issued|set|shake|signed|were/VBD
6	4	0.75	,|:/, TARGET and/CC HOOK agree|agreed|are|began|commit|concluded|decided|grew|have|issued|shake|signed|were/VBD
7	4	0.75	,|:/, TARGET and/CC HOOK agree|agreed|are|commit|concluded|decided|grew|have|issued|shake|signed|were/VBP
8	4	0.75	,|:/, TARGET and/CC HOOK agree|agreed|are|commit|concluded|decided|have|issued|signed|were/VBP
9	5	0.80	,|:/, HOOK and/CC TARGET addressed|agreed|are|attempted|began|concluded|confirm|decided|engaged|enjoyed|established|faced|formalized|fought|had|hailed|have|held|issued|offer|opened|propose|put|realized|restored|resumed|said|say|shake|signed|started|took|were/VBD
10	4	1.00	,|:/, HOOK and/CC TARGET addressed|agreed|are|concluded|confirm|engaged|enjoyed|established|faced|formalized|hailed|held|issued|offer|opened|put|realized|restored|resumed|say|shake|signed|started|took|were/VBD
11	4	0.75	,|:/, HOOK and/CC TARGET agreed|are|attempted|began|decided|fought|had|have|propose|said|signed|were/VBD
12	4	0.75	,|:/, HOOK and/CC TARGET agreed|are|began|fought|had|have|said|signed|were/VBP

(102 more rules)

Table 4.24: Rules after pruning for the relation Country/Location-Borders.

No.	Match	Prec.	Rule
1	2	1.00	A/DT new/JJ official/NN flag/NN of/IN HOOK was/VBD adopted/VBN on/IN date/entity by/IN the/DT TARGET legislature/NN [EOS]/[EOS]
2	2	1.00	in|of/IN HOOK is|was/VBZ */* for|on|with/IN */* by|in|with/IN an|the/DT TARGET address|government|legislature/NN

Table 4.25: Rules after pruning for the relation Country-Inhabitant.

No.	Match	Prec.	Rule
1	4	1.00	[BOS]/[BOS] */* date/entity HOOK TARGET 's/POS
2	2	1.00	[BOS]/[BOS] Based/VBN in/IN TARGET http/NN :/, www.bargainhunter.co.zw/NN person/entity on/IN HOOK Article/NNP about/IN person/entity from/IN the/DT
3	2	1.00	[BOS]/[BOS] From/IN HOOK organisation/entity location/entity http/NN :/, www.botswana-online.com/NN TARGET person/entity by/IN Vishvas/NNP person/entity ,/,
4	4	1.00	they/PRP are/VBP time_expr/entity prevalent/JJ throughout/IN HOOK ,/, including/VBG TARGET [EOS]/[EOS]

Table 4.26: Rules after pruning for the relation Country/Area-Continent.

Chapter 5

Conclusions

This work explores: (1) the research on NLP techniques to automatically extract semantics from unrestricted text; (2) the implementation of a prototype to perform these tasks; and (3) the execution and evaluation of the prototype through case experiments.

The literature in Word Sense Disambiguation, Named Entity Recognition and Relationships Extraction outlines many advantages and disadvantages of current techniques, and has served as basis to take many design decisions for the architecture of the system, for the modules implementation and for the settings to be tested. The system designed improves existing NLP methods and adds a novel approach on learning and generalisation of patterns for information extraction.

Also several knowledge sources have been discussed, from dictionaries to lexico semantic databases, from ontologies to free text. For this work, a hybrid approach has been adopted, exploiting at a time the ample availability of free text within the World Wide Web, the narrower vocabulary but accurate knowledge in WordNet and the semi-structured encyclopedic knowledge contained in Wikipedia.

5.1 Contributions

The main contribution of this work is the proposal of new methods to extract knowledge from unstructured text and a system architecture using the web, WordNet and Wikipedia that applies those methods.

Compared with related work using also Wikipedia to extract relations from unstructured text [Herbelot and Copestake, 2006, Suchanek et al., 2006, Nguyen et al., 2007a,b, Blohm and Cimiano, 2007], the methods in this work present the following differences:

- We explore a novel approach combining surface patterns and rote extractors, and uses both WordNet and the WWW to aid the pattern pruning.

- We require a simpler level of natural language processing based on part-of-speech-tagging and NP-chunking, while most related work uses deeper syntax analysis. The use of deeper

syntax can be interesting to enrich the patterns, but (a) restricts the possibilities to port the methods to other languages and (b) involves a higher computational load. The most complex experiment carried out here took two months to execute, for the rote extractor testing, with a Pentium IV laptop computer of 1 GHz and 2 GB RAM, over 19 types of relations and roughly 20000 Wikipedia entries.

Herbelot and Copestake [2006] focus on only one relation (hyponymy) and apply deep parsing. Suchanek et al. [2006] explore three types of relations with a dependency parser[1]. Nguyen et al. [2007a] study 13 relations also with deep parsing, evaluating over roughly 6000 Wikipedia entries (sizing approximately 1/3 in entries number compared to our test corpus). Reversely, Blohm and Cimiano [2007] study 7 relations with an approach that uses Wikipedia and the WWW as data source using only lexical patterns, but with a larger data set sizing approximately 250000 Wikipedia entries (more than 12 times our 20000 entries).

In all these works, there seems to be a trade-off between the complexity of the NLP processing and the quantity of data that was processed: the more complex the process, the smaller the data sets and the number of explored relations. By limiting the level of language analysis to part-of-speech, our approach uses a middle NLP complexity, between the simpler patterns of Blohm and Cimiano [2007] and the approaches using syntax, having the advantage of studying a varied set of relations using a rather large Wikipedia corpus.

In more detail, the following contributions to specific areas have been achieved:

1. Concerning Word-Sense Disambiguation of encyclopedic entries this is, to the best of our knowledge, the first reported work attempting to link Wikipedia encyclopedic entries with WordNet synsets. The procedure proposed is general enough to be applied to other corpora or lexico-semantic networks. Some take-aways from these experiments are:

   (a) The proposed Word Sense Disambiguation algorithm, mainly based on combining the Vector Space Model with the WordNet taxonomy, has been successfully applied to the task. The tests show significantly better performance than the baselines. The best results for this experiment resulted for stemmed words, using tf·idf as weight function and the dot product as similarity metric.

   (b) The approach has been tested with WordNet 1.7 and the Simple English Wikipedia. 1229 WordNet glosses have been extended with encyclopedic entries.

   (c) It has been shown that, for the disambiguation, it is possible to reach accuracy rates as high as 92% (98% for monosemous words and 87% for polysemous words). Interestingly, this result is much higher than current state-of-the-art techniques for general Word Sense Disambiguation of words inside a text, using WordNet as the repository of senses: 73% accuracy in latest SENSEVAL [Mihalcea et al., 2004a] and 89% F-score [Agirre and Lopez de Lacalle, 2007, Hawker, 2007, Cai et al., 2007] in latest

[1] The main author works afterwards in YAGO, extracting relations on a different, large-scale approach based on structured data.

SEMEVAL, indicating that enclyclopedic text is easier to disambiguate: the tests in the present work compare gloss definitions from WordNet with encyclopedic definitions from Wikipedia. As far as it consists in contrasting two definitions, there might be many common words between the encyclopedic entry for the polysemous word and the gloss of the right candidate sense, which results in a high similarity metric. This does not necessary happen when contrasting the glosses with a context obtained from a free-text source.

2. Concerning the extraction of semantic relationships, a new approach based on the automatic acquisition and use of lexicosyntactic patterns has been proposed. Currently, no applicable test benchmarks are available. SEMEVAL-2007 in its Task 4 presented a test bench for relations classification, but it was focused to nominals excluding named entities, hence being not applicable to our work. Although a comparison of works based only in published results is not rigorous, the figures on precision are given as an illustration of findings.

   Some of the conclusions that can be drawn from this contribution are the following:

   (a) A new algorithm for generalising lexical patterns has been defined, implemented and evaluated. It is based on the edit distance algorithm, which has been modified to take into account the part-of-speech tags of the words. This algorithm is fully automatic, as it requires no human supervision for the generalisation.

   (b) For taxonomic and part-of relations, the set of patterns has been found automatically from WordNet and the Wikipedia entries. It was possible to extract new relations for each of the four relationships: hypernymy, hyponymy, meronymy and holonymy. More than 2600 new related pairs have been provided. The precision of the generated patterns ranges between 60% and 65%[2], similar to that of patterns written *by hand* (although they are not fully comparable, as the experimental settings differ) [Kietz et al., 2000, Cimiano et al., 2004b, Berland and Charniak, 1999, Finkelstein-Landau and Morin, 1999]. Herbelot and Copestake [2006] obtained 92% precision using automatic learning for hyponymy on a biology specific corpus, while our experiments are over a generic corpus. Using topic corpora can enhance very much the precision, as illustrated with our experiment on the *football* topic corpus that raised the precision of the *football team-player* relation from 7% to 69.3%. Nevertheless, restricting pattern learning to topic corpora rises a problem on data sparseness, also as pointed out by other authors [Blohm and Cimiano, 2007].

   (c) For general non taxonomic relations the patterns have been learnt from free text in the World Wide Web, using a rote extractor-based pruning. The pruning phase has been enhanced with a novel automatic precision calculation better fit for an information extraction task than the traditional approach, developed for a question-answering task. More than 16000 related pairs were extracted. The precision depends on the type of

[2]Experiment with threshold 3.

relation: people and their birth and death places/dates proved to be easily extracted, summing up to almost 13000 pairs with accuracies between 76% and 96%. The rest of pairs are mainly extracted by relations like author-work and region-borders, with somewhat lower precisions at 35-44%. Some relations (based on soccer, cinema, painting and geographical data) resulted in a poorer capability to extract a significant quantity of pairs, although part of the patterns obtained for them have also very high accuracies, 60 to 100%. The differences in precision and the apparent easiness to extract relations based on biographical data are in line with the results obtained by other authors extracting general relations, e.g. Suchanek et al. [2006] reaches a 74% precision for *birth-date*. Nguyen et al. [2007b] publish an overall precision of 29% in their experiments over 13 different relations, including some of those that we found easier to extract like *birth-date* or *birth-place* but also other more difficult such as *location*. These types are also within the set of 7 relations studied by Blohm and Cimiano [2007], who obtain an overall precision of almost 60%.

3. It is worthy to mention that part of the work served to extend the NLP toolkit "The Wraetlic Tools", enhanced now with a Word-Sense Disambiguation module and a Named Entity Recogniser based on sure-fire rules.

Twelve publications are within the scope of this thesis, including one journal publication, nine international conferences communications and two book chapters. All the new contributions and the related experimental work, as well as part of the literature review, are published as follows:

- The experiments in Word Sense Disambiguation extending Lesk's algorithm with WordNet classes and applying the method to disambiguate Wikipedia entries (Section 4.4) are presented in [Ruiz-Casado et al., 2005a].
- The taxonomic and *part-of* relations extraction from Wikipedia by means of the WordNet data and the novel patterns generalisation algorithm (Section 4.5) are published in [Ruiz-Casado et al., 2005b, 2007].
- The non taxonomic relations extraction through the modified rote extractor and the new pruning methodology (Section 4.5.3) are published in [Alfonseca et al., 2006a,c].
- The experiments with the linguistic tools and improvements (Sections 4.2 and 4.3) are presented in [Alfonseca et al., 2006b, Alfonseca and Ruiz-Casado, 2005, Ruiz-Casado et al., 2005c];
- Literature review-related publications can be found in [Ruiz-Casado et al., 2008a, 2004].
- Finally, we published our explorations in the applications of the metods presented here to semantic environments in the scope of the Semantic Technologies community (Section 4.6), in [Ruiz-Casado et al., 2006, 2008b].

These publications have rendered 89 citations to date[3].

[3]Source: Google Scholar, excluding self-citations, July 2009

5.2 Future Work

The proposed methods can be further enhanced in several directions, such as:

1. A more accurate Named Entity Recogniser would substantially improve the results, as in overall, the failures in the Entity tagger were the main source of error. Syntactic features can also be useful, specially for the detection of complex entities like authored works (books, paintings or films titles, for example).

2. The possibility of extending the lexicosyntactic patterns with deeper syntactic information is worth to explore, but the computational cost should be evaluated. Although related work using deep parsing did not show an advantage in precision, the availability of syntactic features might help improving the pruning phase: currently, many patterns are discarded because the wildcards make them match lexical contexts that hold the relation on focus but also other relations. This procedure discards patterns that are valid, but polysemous. The syntactic information would provide further criteria to decide on the validity of polysemous, highly generalised patterns.

3. The rote extractor approach would highly benefit from the use of a larger training corpus downloaded from the Web and a larger learning corpus from Wikipedia. This would provide a larger set of patterns for each relation and a better pruning quality, which would expectedly improve precision and recall.

4. The improvement on precision when using corpora restricted to a certain theme is worth being noted. Small size topic corpora can rise a problem of data sparseness if they are used solely for learning patterns, i.e. they might not contain enough examples of language use for a certain relation. Nevertheless, they might be useful to disambiguate too general patterns (e.g. **/NNP's */NNP* in *New York's Chrysler Building* and *Barcelona's Etoo*) in the pruning phase.

5. Diverse tasks have a straightforward opportunity to benefit from the methods here developed: the Wikipedia entries that have been linked to WordNet concepts can be used for enriching the glosses and aid in WordNet-based NLP algorithms. Also the relations for which a good recall and precision was reached provide several thousands of related pairs that can be used in applications such as the automatic generation of Wikipedia categories lists or to drive further the efforts to build up a Semantic Wikipedia.

Amongst the many research lines in Semantic Technologies, such as ontology definition standards and tools, ontology exploitation tools (parsers, inference engines, semantic portals), ontology management, Semantic Web services technologies, to mention a few, NLP technologies have been identified as a useful tool for the automatic extraction of semantics, which has multiple applications in ontology enrichment and semantic annotation. This works aims to contribute to fill the gap between unstructured and structured text, and is part of the recent stream of efforts in

developing NLP-based tools that use semantic technologies to tackle the knowledge acquisition bottleneck.

Appendix A

Introducción

A.1 Tecnologías semánticas

El Procesamiento de Lenguaje Natural (PLN) es un área asentada que se halla entre la Inteligencia Artificial y la Lingüística. Ya a mediados del siglo pasado, los lingüistas computacionales comenzaron a estudiar el procesamiento automático de datos textuales para traducción automática. El término "Lingüística Computacional" se acuñó en los años 60, y se continuó investigando en temas como análisis sintácticos y desambiguación de sentidos de palabras.

Estos primeros trabajos estaban muy limitados por las capacidades computacionales de entonces y la escasa disponibilidad de texto en formato electrónico. La situación actual del área es bastante diferente, y la tecnología ha evolucionado considerablemente después de casi seis décadas de investigación. No obstante, muchos de los problemas para el procesamiento automático identificados entonces aún están lejos de ser resueltos completamente.

En este tiempo no ha dejado de crecer el número de aplicaciones del PLN, incluyendo la traducción automática, la extracción de información, la recuperación de información, la generación automática de resúmenes o el análisis de sentimiento. En particular, la extracción de información trata de obtener datos estructurados a partir de datos sin estructura, como es el caso del texto en lenguaje natural.

A partir de los años 90, la World Wide Web (WWW) supuso una revolución en la disponibiliad de textos en formato electrónico. Desde el primer proyecto presentado por Tim Berners Lee en 1989 ha habido muchos avances en tecnología web. Estos incluyen el desarrollo del lenguaje HTML y del protocolo HTTP, que facilitaron la creación y transferencia de contenidos que integran texto y otros medios; la generación dinámica de páginas web; el desarrollo de navegadores web; los motores de búsqueda ; y la adaptación a perfiles de usuario, entre otros. El tamaño estimado de la web es de más de 25×10^9 documentos indexables[1]. La web profunda, que incluye la información a partir de la cual se generan las páginas dinámicas, se estima que es aún mayor que el volumen total de información impresa en el mundo [Castells, 2003].

[1] http://worldwidewebsize.com, Febrero de 2009.

La disponibilidad de tal cantidad de contenidos ha convertido la World Wide Web en un recurso universal de información, dado que prácticamente toda la infromación general puede encontrarse en ella, aunque no siempre la calidad de los datos satisfaga las expectativas del usuario. El espectro de datos en términos de calidad del lenguaje y estructura del contenido varía mucho: desde recursos de conocimiento on-line gratuitos, a veces construidos colaborativamente y con una buena estructura como Wikipedia, que compite en calidad con conocidas enciclopedias desarrolladas por profesionales, a blogs, foros y redes sociales donde la información es menos estructurada y los contenidos son subjetivos, a veces inexactos, y el lenguaje en uso es coloquial o incluso argot. La enorme cantidad de contenidos y la mezcla de fuentes de información con diferentes calidades have que los resultados devueltos por motores de búsqueda a menudo incluyan información irrelevante junto con la información deseada.

Berners-Lee et al. [2001] y Fensel et al. [2002] mencionan muchos ejemplos donde el crecimiento exponencial de la World Wide Web y la inmensa disponibilidad de datos hacen de las tareas de búsqueda, recuperación de datos y mantenimiento de contenidos un trabajo difícil cuando se han de realizar manualmente. A finales de la década pasada emergió la visión de la Web Semántica. Desde que Berners Lee introdujo esta visión, una comunidad importante de investigación en universidades, gobiernos y empresas privadas sumaron esfuerzos en la búsqueda de una Web donde el contenido estaría dotado de una estructura subyacente, organizada de tal manera que la información irrelevante para la tarea deseada (búsqueda y recuperación, gestión de contenido, etc.) pudiera ser descartada y solamente la información relevante fuera seleccionada, permitiendo la automatización del procesamiento de información. Al realizar una búsqueda, por ejemplo, agentes o procedimientos automáticos negociarían para compartir información y seleccionar las mejores respuestas. Ellos gestionarían el contenido web con una supervisión humana reducida al mínimo. Aunque se han dado algunos pasos en esa dirección, la tarea es compleja, y una de las dificultades reconocidas que impide la automatización completa de estos procesos es el hecho de que los contenidos de la web se presentan principalmente en lenguaje natural, cuyas ambigüedades son difíciles de procesar por una máquina [Ding et al., 2002].

El lenguaje natural que usamos los humanos es ambiguo. Las lenguas humanas contienen palabras polisémicas en sus vocabularios, lo cual significa que una palabra se puede utilizar para representar más de un concepto. Por ejemplo, la palabra *bank* en inglés tiene doce significados diferentes en el diccionario Merriam-Webster on-line:

- El terreno que bordea un lago, río o mar (...).
- Una cuesta empinada (...).
- La inclinación lateral de una superficie o un vehículo al tomar una curva.
- Una pieza protectora o amortiguadora.
- Establecimiento para la custodia, préstamo, cambio o emisión de dinero (...).
- La mesa o lugar de negocio de un cambista de moneda.
- La persona que conduce una casa de apuestas.
- Un suministro de algo que se mantiene en reserva (...).
- Un lugar donde algo se mantiene disponible (...).

- Un grupo o serie de objetos colocados en fila.
- Una de las divisiones inferiores horizontales, y generalmente secundarias, de un titular.

Hay palabras más polisémicas que otras. Por ejemplo, el nombre *mano* tiene catorce significados en el lexicón WordNet, mientras que el nombre *uña* tiene sólo tres sentidos. Algunas expresiones son monosémicas en un diccionario, como *elefante indio*, que sólo tiene un sentido en WordNet. Algunos expertos en semántica postulan que las palabras más polisémicas representan conceptos generales, muy comunes, mientras que los términos más raros suelen ser monosémicos [Magnus, 2001]. En cualquier caso, la polisemia dificulta la recuperación de información, y puede comprobarse que, para muchas consultas polisémicas, los motores de búsqueda producen resultados que se refieren a sentidos diferentes de las palabras.

Para evitar las ambigüedades y hacer explícito el significado de los datos, y por tanto hacerlos procesables por una máquina, una práctica común es la anotación del sentido de ciertas palabras, páginas o recursos web usando un repositorio de sentidos, que puede ser, por ejemplo, el formalismo de representación del conocimiento llamado *ontología*. Este formalismo se explica con detalle en el Capítulo 2, y contiene las palabras que representan un cierto dominio, incluyendo los diversos sentidos de aquellas que sean polisémicas, así como relaciones entre las palabras. Algunas relaciones comunes son las relaciones de subclase, o las relaciones *parte-de*. Por ejemplo, si tenemos los conceptos *perro* y *mamífero*, podemos decir que el primero es una subclase del segundo. De la misma forma, hay una relación *parte-de* entre *cola* y *perro*. Los sentidos de las palabras polisémicas se pueden distinguir, por ejemplo, a través de sus diferentes relaciones con otras palabras. Siguiendo el ejemplo anterior de la palabra *bank* en inglés, el sentido de terreno junto al río se puede diferenciar de la entidad financiera en la ontología. Aunque son conceptos que comparten la misma cadena léxica, el primer sentido es un subclase de *formación geológica* y el otro sentido es una subclase de *institución, organización*. La anotación en una página web de la palabra *bank* apuntará a uno de sus posibles sentidos representados en la ontología.

Actualmente la visión de la Web Semántica es más amplia y relajada, dando lugar a las llamadas Tecnologías Semánticas. Se reconoce la complejidad de producir una web soportada por un contenido completamente estructurado y la comunidad investigadora, en lugar de perseguir una web completamente estructurada que permitiría una entera automatización, busca hacer disponibles técnicas en semántica que permitan una mejora continua en las tareas de búsqueda, recuperación de datos, extracción de contenido y anotación, entre otras. Muchas de las tecnologías semánticas provienen de aquellos esfuerzos iniciados durante el siglo pasado y se desarrollan hoy en día desde diversas comunidades, uniendo ideas de la Web Semántica, de Lingüística Computacional y Aprendizaje Automático con el fin común de ofrecer un avance incremental en el enriqueciemiento semántico de datos que haga nuestra web más fácil de procesar. Se está extendiendo la presencia de técnicas semánticas que ayudan en la gestión del conocimiento y ya están en uso en nuestra web actual, por ejemplo a través de los principales motores de búsqueda como Yahoo, Google o Bing, que ofrecen un procesamiento (limitado) de consultas en lenguaje natural.

A.2 Motivación y Objetivos

Una de las aplicaciones de las tecnologías semánticas es la anotación de contenidos hipermedia con información semántica. Bajo esta óptica, los conceptos relevantes que aparecen en las páginas web estarían anotados con el identificador de un concepto de una ontología, de manera que el texto simple estaría vinculado al conocimiento estructurado. La asignación de dichas anotaciones recibe el nombre de *anotación semántica*. Mediante la anotación de contenidos web se marca explícitamente el significado de palabras importantes de una página web, haciendo así que el texto pueda ser procesado automáticamente por una máquina sin ambigüedades de significado.

En la práctica, construir y mantener de forma manual ontologías, así como anotar manualmente las páginas web, puede ser una tarea costosa en cuanto a tiempo y recursos, siendo esta dificultad una conocida restricción para el objetivo de enriquecer la web con contenido semántico. Muchos autores han señalado la necesidad de procedimientos automáticos o semi-automáticos que transformen texto sin estructura en conocimiento estructurado [Contreras et al., 2003, Contreras, 2004, Gómez-Pérez et al., 2003, Kietz et al., 2000, Maedche and Volz, 2001, Maedche and Staab, 2000].

Dirigiéndose al problema de enriquecer textos hipermedia mediante anotaciones estructuradas, este trabajo se enfoca en la semi-automatización de la identificación de conocimiento para el enriquecimiento de ontologías y anotación semántica. De entre las diferentes aproximaciones, este trabajo se centra en la aplicación y mejora de técnicas de PLN para tal fin.

El objetivo principal de este estudio es el diseño e implementación de técnicas de PLN para ayudar en la anotación de textos y extensión de ontologías. Para ello, se realizan las siguientes tareas:

1. La mejora de técnicas de desambiguación de sentido de palabras para identificar el sentido con que una palabra se utiliza en un contexto. Por ejemplo, dadas las frases

   (4) Después de pasear me senté en un banco.

   (5) El préstamo que me concedieron en el banco me va a venir muy bien.

   podemos adivinar el significado de *banco* en cada una de ellas a partir de la evidencia proporcionada por las otras palabras en sus contextos, por ejemplo *senté* en la primera oración y *préstamo* en la segunda.

2. La mejora de los sistemas de extensión de ontologías aprendiendo nuevas relaciones entre los conceptos del texto. Por ejemplo, dada la frase

   (6) El perro meneó su cola.

   indica que *cola* es una parte de *perro*, debido a la presencia del pronombre posesivo *su*.

Para abordar estos problemas, se proponen aquí varios métodos, principalmente basados en (1) el uso de WordNet como fuente lexicosemántica externa, (2) métodos estadísticos basados en el modelo de espacio vectorial para la desambiguación, (3) extracción y generalización de

patrones para la detección de relaciones entre los términos, y (4) métodos de aprendizaje basados en extracción de repeticiones para categorizar y limpiar las relaciones obtenidas.

Estos métodos se han estudiado para estudiar su habilidad de alcanzar los objetivos mencionados. Se presenta además un experimento utilizando el vocabulario controlado de una enciclopedia on-line, la Wikipedia. En él, se desambiguan las entradas de la Wikipedia utilizando los sentidos del lexicón, y se enriquece éste con las definiciones enciclopédicas y las nuevas relaciones halladas en las entradas.

Appendix B

Conclusiones

Este trabajo explora: (1) la investigación de técnicas de PLN para la extracción de semántica a partir de texto no restringido; (2) la implementación de un prototipo para realizar estas tareas; (3) la ejecución y evaluación del prototipo mediante experimentos.

La revisión de literatura en desambiguación de sentidos, reconocimiento de entidades y extracción de relaciones resalta las ventajas y desventajas de las técnicas existentes, y ha servido como base para tomar muchas decisiones acerca del diseño de la arquitectura propueta, de la implementación de los diferentes módulos, y del diseño de la evaluación. El sistema mejora los métodos de PLN existentes y contribuye una técnica nueva de aprendizaje y generalización de patrones para extracción de información.

Además, se discuten diversas fuentes de conocimiento, de diccionarios a bases de datos léxico-semánticas, de ontologías a texto libre. Para este trabajo se ha adoptado una aproximación híbrida, explotando a un mismo tiempo la amplia disponibilidad de texto libre en la web, el vocabulario algo más restringido pero más fiable de WordNet, y el conocimiento semi-estructurado contenido en la Wikipedia.

B.1 Contribuciones

La principal contribución de este trabajo es la propuesta de nuevos métodos para extraer conocimiento a partir de texto no estructurado, y una arquitectura de sistema que utiliza la web, la Wikipedia y WordNet para aplicar estos métodos.

Comparada con el trabajo relacionado que utiliza la Wikipedia para extraer relaciones de texto no estructurado, [Herbelot and Copestake, 2006, Suchanek et al., 2006, Nguyen et al., 2007a,b, Blohm and Cimiano, 2007], los métodos de este trabajo tienen las siguientes diferencias:

- Exploramos una técnica nueva que combina patrones superficiales y extratores de repeticiones, y utiliza tanto WordNet como la WWW para ayuda en la limpieza de los patrones.
- Requerimos niveles de procesamiento de lenguaje natural más simples, basados en anotación de partes del lenguaje y detectores de sintagmas nominales no recursivos. En con-

traste, la mayor parte del trabajo relacionado usa análisis sintácticos profundos. El uso de éstos puede ser interesante para enriquecer los patrones, pero (a) restringe las posibilidades de portar los métodos a otros idiomas, y (b) conlleva una carga computacional mayor. El experimento más complejo llevado a cabo en este trabajo llevó dos meses de tiempo de ejecución, para el extractor de repeticiones, en un ordenador portátil Pentium IV de 1 Ghz y 2 GB de memoria RAM, con un conjunto de 19 tipos de relaciones y 20.000 entradas de la Wikipedia. A pesar de la existencia de clústers de procesamiento paralelos que serían de gran utilidad para este trabajo, dado el tamaño total de los corpus actuales de la web y la Wikipedia, es útil disponer de procedimientos menos costosos.

Herbelot and Copestake [2006] ponen la atención en sólo una relación (hiponimia), y aplican análisis sintácticos profundos. Suchanek et al. [2006] explora tres tipos de relaciones con un analizador de dependencias sintácticas[1]. Nguyen et al. [2007a] estudian 13 relaciones también con análisis sintácticos profundos, evaluado sobre 6.000 entradas de la Wikipedia (aproximadamente un tercio de las entradas de nuestro corpus de test). Por otra parte, Blohm and Cimiano [2007] estudian siete relaciones con una aproximación que utiliza la Wikipedia y la web como fuente de datos utilizando sólo patrones léxicos, pero con un conjunto de datos de 250.000 entradas de la Wikipedia (más de doce veces el tamaño de nuestro corpus).

En todos estos trabajos parece haber un compromiso entre la complejidad de las aplicaciones de PLN y la cantidad de datos procesada: cuanto más complejo el proceso, menores los conjuntos de datos y el número de relaciones exploradas. Limitando el nivel de análisis de lenguaje a anotaciones de partes del lenguaje, nuestra aproximación utiliza una complejidad de PLN intermedia entre los patrones más simples de Blohm and Cimiano [2007] y las aproximaciones más complejas utilizando sintaxis profunda, con la ventaja de estudiar un conjunto variado de relaciones utliizando un corpus de la Wikipedia de buen tamaño.

Se han realizado las siguientes contribuciones a áreas específicas de PLN:

1. En lo que respecta a la desambiguación de sentidos de las entradas enciclopédicas, se trata, hasta donde sabemos, del primer trabajo publicado que trata de enlazar el conocimiento enciclopédico de la Wikipedia con los synsets de WordNet. El procedimiento propuesto es lo bastante genérico como para ser aplicado a otros córpora o redes léxico-semánticas. Algunos resultados importantes de estos experimentos son:

   (a) El algoritmo de desambiguación propuesto, basado principalmente en combinar un modelo de espacio vectorial con la taxonomías de WordNet. La evaluación muestra una precisión significativamente mejor que los baselines. El mejor resultado para este experimento se obtuvo utilizando la forma canónica de las palabras, utilizando tf·idf como función de peso, y el producto escalar como métrica de similitud.

[1] El autor principal trabajó también en YAGO, extrayendo relaciones usando otro procedimiento diferente basado en datos estructurados.

(b) Este método se ha probado utilizando WordNet 1.7 y la Wikipedia en inglés sencillo. 1.229 definiciones de WordNet se han extendido con entradas enciclopédicas.

(c) Se ha demostrado que, para la desambiguación, es posible alcanzar tasas de precisión tan altas como el 92% (87% para las palabras polisémicas y 98% para las monosémicas). Es interesante resaltar que este resultado es mucho mayor que el estado del arte actual desambiguando palabras de un texto utilizando los sentidos de WordNet (73% en el último SENSEVAL, y 89% de F-score [Agirre and Lopez de Lacalle, 2007, Hawker, 2007, Cai et al., 2007] en el último SEMEVAL. Esto es debido a que los textos enciclopédicos son más fáciles de desambiguar, ya que estamos comparando definiciones de WordNet con definiciones de la Wikipedia. Es de esperar que dos definiciones diferentes del mismo concepto compartirán bastantes palabras, lo que resultará en una similitud mayor. Esto no ocurre necesariamente entre una definición y un contexto extraído de texto libre.

2. En lo que respecta a la extracción de relaciones semánticas, se propone un nuevo procedimiento basado en la adquisión y utilización automática de patrones léxico-sintácticos. En el momento de realizar el trabajo no existía ningún test de evaluación estándar. SEMEVAL-2007, en su tarea número 4, presentó un conjunto de evaluación para clasificación de relaciones, pero estaba enfocado a nombres comunes, excluyendo entidades nombradas, por lo que no era directamente aplicable a este trabajo. Aunque una comparación de trabajos basada sólo en resultados publicados no es rigurosa, se proporcionan los datos de precisión como ilustración de los resultados obtenidos.

   Algunas de las aportaciones de este trabajo son:

   (a) Se ha definido, implementado y evaluado un algoritmo nuevo para generalizar patrones léxicos. Está basado en un algoritmo de distancia de edición, modificado para tener en cuenta las partes del lenguaje de las palabras. El algoritmo es completamente automático, dado que no quiere ninguna supervisión humana.

   (b) Para las relaciones taxonómica y es-parte-de, el conjunto de patrones se ha extraído automáticamente utilizando como punto de partida WordNet y las entradas de la Wikipedia. Ha sido posible extraer nuevas relaciones para hiponima, hiperonimia, meronimia y holonimia. Se proporcionan más de 2.600 nuevos pares relacionados. La precisión de los patrones obtenidos se encuentra entre el 60% y el 65%[2], similar al de patrones escritos a mano (aunque no son completamente comparables, dado que los datos de prueba no son los mismos) [Kietz et al., 2000, Cimiano et al., 2004b, Berland and Charniak, 1999, Finkelstein-Landau and Morin, 1999]. Herbelot and Copestake [2006] obtuvo una precisión del 92% utilizando un sistema para aprender automáticamente relaciones de hiponimia de un corpus de biología, mientras que nuestros experimentos son en un corpus más genérico. Utilizando córpora específicos

[2]Experimento utilizando el umbral 3.

de dominio puede producir mayor precisión, tal como se ilustra con nuestro experimento sobre el corpus de fútbol, para el cual la precisión aumenta del 7% al 69,3% al restringir la aplicación de los patrones a documentos sobre fútbol. Sin embargo, restringir el aprendizaje de patrones a córpora específicos provoca el problema de la falta de datos, tal como ha sido señalado por otros autores [Blohm and Cimiano, 2007].

(c) Para las relaciones genéricas, no taxonónimcas, los patrones se han aprendido de texto libre en la World Wide Web, utilizando un procedimiento de aprendizaje de repeticiones. La fase de limpieza de los patrones obtenidos se ha mejorado con un procedimiento de cálculo de precisión automático, más adecuado para la extracción de información que el procedimiento tradicional que había sido desarrollado para obtención de respuestas. Se han extraído más de 16.000 pares de entidades relacionadas. La precisión depende el tipo de relación, de modo que la fecha y el lugar de nacimiento resultaron ser fáciles de extraer, incluyendo casi 13.000 pares con precisiones entre el 76% y el 96%. El resto de los pares se extrajeron principalmente para relaciones tales como autor-obra y región-frontera, con unas precisiones algo inferiores, en el rango de 35% a 44%. Algunas relaciones, basadas fúbol, cine, pintura o datos geográficos, resultaron ser más difíciles utilizando este procedimiento, aunque algunos de los patrones obtenidos para ellas tienen precisiones muy altas, de 60 a 100%. Las diferencias en precisión y la facilidad aparente de extraer relaciones basadas en datos biográficos están de acuerdo con resultados obtenidos por otros autores, e.g. Suchanek et al. [2006] obtuvo un 74% de precisión para *fecha de nacimiento*. Nguyen et al. [2007b] reporta una precisión del 29% sobre 13 relaciones iferentes, incluyendo algunas más sencillas como *fecha de nacimiento* o *lugar de nacimiento*, pero también otras más difíciles como *lugar*. Estos tipos también están entre el conjunto de siete relaciones estudiadas por Blohm and Cimiano [2007], quien obtuvo una precisión total de casi el 60%.

3. Es digno de mención que parte del trabajo realizado aquí sirvió para extender el toolkit de PLN *wraetlic*, que ahora incluye un módulo de desambiguación de sentidos de palabras y otro de reconocimiento de entidades basado en patrones.

Se han realizado doce publicaciones durante el trabajo de esta tesis, incluyendo un artículo de revista, nueve comunicaciones a congresos internacionales, y dos capítulos de libro. Todas estas contribuciones y el trabajo experimental relacionado, así como parte de la revisión de literatura, se han publicado como sigue:

- Los experimentos de desambiguación de palabras extendiendo el algoritmo de Lesk con clases de WordNet, y aplicando el algoritmo para desambiguar las entradas de la Wikipedia (Sección 4.4) se describen en [Ruiz-Casado et al., 2005a].

- Las relaciones taxonómicas y es-parte-de, extraídas de la Wikipedia mediante el uso de

WordNet y patrones léxico-sintácticos (Sección 4.5) se han publicado en [Ruiz-Casado et al., 2005b, 2007].

- La extracción de relaciones no taxonómicas mediante el extractor de repeticiones y el nuevo procedimiento de evaluación de patrones (Sección 4.5.3) se han publicado en [Alfonseca et al., 2006a,c].

- Los experimentos con las herramientas lingüísticas y las mejoras (Secciones 4.2 y 4.3) se describen en [Alfonseca et al., 2006b, Alfonseca and Ruiz-Casado, 2005, Ruiz-Casado et al., 2005c];

- Publicaciones relacionadas con la revisión de literatura son [Ruiz-Casado et al., 2008a, 2004].

- Finalmente, se han publicado las exploraciones en las aplicaciones de estos métodos para entornos semánticos, dentro de la comunidad de web semántica (Sección 4.6), en [Ruiz-Casado et al., 2006, 2008b].

Estas publicaciones han dado lugar a 89 citas hasta la fecha[3].

B.2 Trabajo futuro

Los métodos propuestos se pueden mejorar siguiendo varias direcciones, incluyendo las siguientes:

1. Un reconocedor de entidades nombradas más exacto mejoraría substancialmente los resultados, dado que errores en la anotación de éstas fue la principal causa de errores del sistema completo. La sintaxis podría ser útil especialmente para la detección de nombres de entidades complejas, como obras (titulos de libros, cuadros o películas, por ejemplo).

2. Vale la pena explorar la extensión de los patrones léxico-sintácticos con información sintáctica más profunda, aunque se debería evaluar el coste computacional. Aunque el trabajo relacionado que utiliza análisis sintácticos profundos no muestra una mayor precisión, la disponibilidad de éstos podría ayudar a mejorar la fase de limpieza de patrones: actualmente muchos patrones se descartan porque los comodines (wildcards) hacen que los patrones extraigan pares de otras relaciones diferentes. Este procedimiento descarta patrones que son válidos, pero polisémicos. La información sintáctica proporcionaría más criterios para decidir acerca de la validez de los patrones polisémicos y demasiado generalizados.

3. El extractor de repeticiones se beneficiaría del uso de un corpus de entrenamiento de mayor tamaño descargado de la web, y de un corpus mayor de la Wikipedia. Esto daría lugar a un conjunto de patrones de mayor tamaño y mejor evaluado.

[3] Source: Google Scholar, excluyendo autocitas, Julio de 2009

4. Las mejoras en precisión cuando se utilizó un corpus restringido a un tema en particular es digno de mención también. Córpora específicos, pero pequeños, pueden dar lugar a un problema de escasez de datos si se utilizan sólo para aprendizaje de patrones, es decir, podrían no centener bastantes ejemplos de uso del lenguaje para una cierta relación. Sin embargo, podrían ser útiles para desambiguar patrones muy generales. Clasificar el tema de un documento antes de aplicar un patrón polisémico podría ayudar en la obtención de resultados más exactos.

5. Varias tareas diferentes se pueden beneficiar directamente de los métodos aquí desarrollados: las entradas de la Wikipedia que se han enlazado a los conceptos de WordNet se pueden usar para enriquecer las definiciones y ayudar a otros algoritmos basados en WordNet. Asímismo, las relaciones para las que se ha obtenido una buena precisión y cobertura proporcionan miles de pares relacionados que se pueden aplicar, por ejemplo, en la generación automática de listas de categorías de la Wikipedia, o para avanzar en los esfuerzos de generar una Wikipedia semántica.

Entre las muchas líneas de investigación en tecnologías semánticas, tales como la definición de estándares y herramientas de ontologías, o las herramientas para utilizar y gestionar ontologías, las tecnologías de PLN se han identificado como muy útiles para la extracción automática de semántica, con múltiples aplicaciones en el enriquecimiento de ontologías y la anotación semántica. Este trabajo trata de contribuir en acortar la distancia entre el texto estructurado y el no estructurado, y es parte de una corriente reciente de esfuerzos en el desarrollo de herramientas basadas en PLN que utilizan tecnologías semánticas para tratar el cuello de botella de la adquisición de conocimiento.

Bibliography

S. Abney. Parsing by chunks. In *Robert C. Berwick, Steven P. Abney, and Carol Tenny, editors, Principle-Based Parsing: Computation and Psycholinguistics*, pages 257–278, Boston, 1991. Kluwer Academic Publishers.

E. Agichtein and L. Gravano. Snowball: extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM New York, NY, USA, 2000.

E. Agirre, O. Ansa, E. Hovy, and D. Martinez. Enriching very large ontologies using the www. In *Ontology Learning Workshop, ECAI*, Berlin, Germany, 2000.

E. Agirre, O. Ansa, D. Martínez, and E. Hovy. Enriching wordnet concepts with topic signatures. In *Proceedings of the NAACL workshop on WordNet and Other lexical Resources: Applications, Extensions and Customizations*, Pittsburg, 2001.

Eneko Agirre and Oier Lopez de Lacalle. Ubc-alm: Combining k-nn with svd for wsd. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 342–345, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

E. Alfonseca and S. Manandhar. Distinguishing instances and concepts in wordnet. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, 2002a.

E. Alfonseca and S. Manandhar. Improving an ontology refinement method with hyponymy patterns. In *Language Resources and Evaluation (LREC-2002)*, Las Palmas, 2002b.

E. Alfonseca and S. Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, 2002c.

E. Alfonseca and P. Rodríguez. Lecturing notes of the phd course in natural lnaguage processing, 2003.

E. Alfonseca and M. Ruiz-Casado. Learning sure-fire rules for named entities recognition. In *Proceedings of the International Workshop in Text Mining Research, Practice and Opportunities, in conjunction with RANLP conference*, Borovets, Bulgary., 2005.

E. Alfonseca, P. Castells, M. Okumura, and M. Ruiz-Casado. A rote extractor with edit distance-based generalisation and multi-corpora precision calculation. In *21st. International COnference on Computational Linguistics ad 44th. Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, Sydney, Australia, 2006a.

E. Alfonseca, A. Moreno-Sandoval, J.M. Guirao, and M. Ruiz-Casado. The wraetlic nlp suite. In *Proceedings of the Language Resources and Evaluation Conference (LREC-2006)*, 2006b.

E. Alfonseca, M. Ruiz-Casado, M. Okumura, and P. Castells. Towards large-scale non-taxonomic relations extraction: Estimating the precision of rote extractors. In *2nd Workshop on Ontology Learning and Population: Bridging the gap between Text and Knowledge, in the (COLING-ACL 2006)*, Sydney, Australia, 2006c.

I. Androutsopoulos and M. Aretoulaki. Natural language interaction. In *Ruslan Mitkov (ed.) The Oxford Handbook of Computational Linguistics*, pages 136–156. Oxford University Press, 2003.

M. Arevalo, M. Civit, and M.A. Marti. Mice: A module for named entities recognition and classification. *International Journal of Corpus Linguistics*, 9(1):53–68, 2004.

J. Artiles, J. Gonzalo, and F. Verdejo. A testbed for people searching strategies in the www. In *Proceedings of SIGIR-2005*, pages 569–570, 2005.

B. Auer and J. Lehman. What have Innsbruck and Leipzig in common? In *Proceedings of the ESWC 2007*. Springer, 2007.

B. Auer, C. Bizer, J. Lehman, g. Kobilarov, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of the ISWC/ASWC 2007*, 2007.

N. Aussenac-Gilles, B. Biébow, and S. Szulman. Corpus analysis for conceptual modelling. In *Proceedings of the 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW)*, Juan-les-Pins, France, 2000.

B. Bachimont, A. Isaac, and R. Troncy. Semantic commitment for designing ontologies: a proposal. In *Knowledge Engineering and Knowledge Management*, volume 2473 of *Lecture Notes in Artificial Intelligence*, pages 114–121. Springer Verlag, 2002.

A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 1998 International Conference on Computational Linguistics*, 1998.

J. Bateman. Natural language generation: an introduction and open-ended review of the state of the art, 2002. URL `http://www.fb10.uni-bremen.de/anglistik/langpro/webspace/jb/info-pages/nlg/ATG01/ATG01.html`.

Luisa Bentivogli, Pamela Forner, Bernardo Magnini, and Emanuele Pianta. Revising wordnet domains hierarchy: Semantics, coverage, and balancing. In *Proceedings of COLING 2004 Workshop on Multilingual Linguistic Resources*, pages 101–108, Geneva, Switzerland, 2004.

M. Berland and E. Charniak. Finding parts in very large corpora. In *Proceedings of the Association of Computational Linguistic*, 1999.

T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web - a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 284(5):34–43, may 2001.

W.J. Black and A. Vasilakopoulos. Language-independent named entity classification by modified transformation learning and by decision tree induction. In *6th workshop on Computational Language Learning in 19th International Conference on Computational Linguistics*, pages 159–162, Taipei, Taiwan, 2002.

S. Blohm and P. Cimiano. Using the Web to reduce data sparseness in pattern-based Information Extraction. In *Proceedings of the 11th PKDD European COnference 2007*. Springer, 2007.

T. Brants. *TnT - A Statistical Part-of-Speech Tagger*. User manual, 2000.

E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. Word sense disambiguation using statistical methods. In *Proceedings of the Annual Meeting of the ACL*, pages 264–270, 1991.

P. F. Brown, V.J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

P. Buitelaar. CORELEX*: Systematic Polysemy and Underspecification*. Ph.D. Thesis. Brandeis University, Department of Computer Science, 1998.

R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of 11th. COnference of the European Chapter of the Association of Computational Linguistics (EACL06)*, pages 9–16, Trento, Italy, 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 Morristown, NJ, USA, 2005.

J. Burstein and C. Leacock. Automated evaluation of essays and short answers. In *Conference TALN 2000*, 2000.

M. T. Cabré, R. Estopá, and J. Vivaldi. Automatic term detection: a review of current systems. In *Recent advances in computational terminology, volume 2 of Natural Language Processing*, pages 53–87. John Benjamins, 2001.

Jun Fu Cai, Wee Sun Lee, and Yee Whye Teh. Nus-ml:improving word sense disambiguation using topic features. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 249–252, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

D. Calvanese, G. Di Giacomo, and M. Lenzerini. The emerging semantic web. In *Selected Papers from the 1st. International Semantic Web Working Symposium (SWWS)*, pages 201–214, Stanford University, California, 2002. IOS press.

S. A. Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 120–126, 1999.

C. Cardie. A case-based approach to knowledge acquisition for domain-specific sentence analysis. In *Proceedings of the Eleventh National Conference on Artificial INtelligence*, pages 798–803, 1993.

X. Carreras, L. Marquez, and L. Padro. A simple named entity extractor using adaboost. In *Proceedings of the seventh Conference on Natural Language Learning*, pages 152–155, Edmonton, Canada, 2003.

P. Castells. *La Web Semantica. In C. Bravo and M. A. Redondo (Eds.), Sistemas Interactivos y Colaborativos en la Web*, pages 195–212. Ediciones de la Universidad de Castilla-La mancha, 2003.

D. Chakrabarti, D. K. Narayan, P. Pandey, and P. Bhattacharyya. Experiences in building the indo wordnet: A wordnet for hindi. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, january 2002.

Yee Seng Chan, Hwee Tou Ng, and Zhi Zhong. Nus-pt: Exploiting parallel texts for word sense disambiguation in the english all-words tasks. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 253–256, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

E. Charniak. A maximum-entropy-inspired parser. In *ACM International Conference Proceeding Series*, volume 4, pages 132–139, 2000.

N. Chomsky. *Syntactic Structures*. The Hague:Mouton, 1957.

K. Church, W. Gale, P. Hanks, and D. Hindle. *Using Statistics in Lexical Analysis. In U. Zernik (ed.), Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon*, chapter 6, pages 115–164. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1991.

P. Cimiano and J. Volker. Toward large-scale, open-domain and ontology-based named entity classification. In *Proceedings of the Recent Advances in Natural Language Processing Conference, RANLP-2005*, Borovets, Bulgary, 2005.

P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *Proceedings of the 13th World Wide Web Conference*, 2004a.

P. Cimiano, A. Hotho, and S. Staab. Clustering concept hierarchies from text. In *Proceedings of the 4th International Conference on Language Resources and Evaluation LREC*, Lisbon, Portugal, 2004b.

P. Cimiano, G. Ludwig, and S. Staab. Gimmeťhe context: Context-driven semantic annotation with c-pankow. In *Proceedings of the 14th. International World Wide Web Conference*, Chiba, Japan, 2005.

J. Colas. Lecturing notes of the phd course in speech processing, 2003.

M. Collins. Head-Driven Statistical Models for Natural Language Parsing. *Computational Linguistics*, 29 (4):589–637, 2003.

J. Contreras. *Incremento del Contenido Crítico de la Web Semántica mediante Poblado Automático de Ontologías*. Ph.D. thesis. Universidad Politécnica de Madrid. Facultad de Informática, 2004.

J. Contreras, R. Benjamins, F. Martin, B. Navarrete, G. Aguado, I. Álvarez, A. Pareja, and R. Plaza. Esperonto deliverable d31: Annotation tools and services, 2003.

A. Copestake. The acquilex lkb: representation issues in semi-automatic acquisition of large lexicons. In *3rd Conference on Applied Natural Language Processing (ANLP-92)*, Trento, Italy, 1992.

A. Culotta and J. Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 423–429, 2004.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. The gate user guide, 2002.

K. G. Dahlgren. *Naive Semantics for Natural Language Understanding*. Kluwer Academic Publishers, Boston, 2000.

H. Daumé III and D. Marcu. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of the joint conference on Human Language Technology and Empirical Methods in Natural Language Processing HLT/EMNLP-2005*, pages 97–104, 2005.

Dmitry Davidov and Ari Rappoport. Classification of semantic relationships between nominals using pattern clusters. In *Proceedings of ACL-08: HLT*, pages 227–235, Columbus, Ohio, June 2008a. Association for Computational Linguistics.

Dmitry Davidov and Ari Rappoport. Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated SAT analogy questions. In *Proceedings of ACL-08: HLT*, pages 692–700, Columbus, Ohio, June 2008b. Association for Computational Linguistics.

M. DeBoni and S. Manandhar. Automated discovery of telic relations in wordnet. In *Proceedings of the 1st International Conference of General WordNet*, Mysore, India, 2002.

Y. Ding, D. Fensel, M. Klein, and B. Omelayenko. The semantic web: Yet another hip? *Data and Knowledge Engineering*, 41(2–3):205–207, 2002.

P. Edmonds and S. Cotton. SENSEVAL-2: Overview. In *Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, 2001.

R. Engels and B. Bremdal. On-to-knowledge deliverable 5: Information extraction. state of the art report, 2000.

A. Faatz and R. Steinmetz. Ontology enrichment with text from the www. In *Proceedings of the 2nd Workshop in Semantic Web Mining. Joint conference ECML-PKDD 2002*, Washington D.C., 2002.

D. Faure and C. Nédellec. A corpus-based conceptual clustering method for verb frames and ontology acquisition. In *LREC workshop on Adapting lexical and corpus resources to sublanguages and applications*, Granada, Spain, 1998.

D. Fensel, C. Bussler, Y. Ding, V. Kartseva, M. Klein, M. Korotkiy, B. Omelayenko, and R. Siebes. Semantic web application areas. In *7th. International Workshop on Application of Natural Language to Information Systems*, Stockholm, Sweden, 2002.

J. Fernández, M. Castillo, G. Rigau, J. Atserias, and J. Turmo. Automatic acquisition of sense examples using exretriever. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 25–28, Lisbon, Portugal, 2004.

M. Finkelstein-Landau and E. Morin. Extracting semantic relationaships between terms: Supervised vs. unsupervised methods. In *Proceedings of the International Workshop on Ontological Engineering on the Global Information Infrastructure*, 1999.

D. Fischer, S. Soderland, J. McCarty, F. Feng, and W. Lehnert. Description of the umass system as used for muc-6. In *Proceedings of the sixth Message Understanding Conference (MUC-6)*, pages 127–140, 1995.

R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. In *Proceedings of the seventh Conference on Natural Language Learning*, pages 168–171, Edmonton, Canada, 2003.

R. Florian, H. Jing, N. Kambhatla, and I. Zitouni. Factorizing complex models: A case study in mention detection. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association of Computationl Linguistics*, pages 473–480, Sydney, Australia, 2006.

D. Freitag and A. McCallum. Information extraction with hmm structures learnt by stochastic optimization. In *Proceedings of the 17th National Conference on Computational Linguistics and 12th Conference on Innovative Applications of Artificial Intelligence*, pages 584–589, 2000.

R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks. University of sheffield: Description of the lasie system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 207–220. Morgan Kauffmann, 1995.

W. A. Gale, K. W. Church, and D. Yarowsky. A method for disambiguating word senses in a lare corpus. *Computers and the Humanities*, 26:415–439, 1993.

B. Ganter and R. Wille. *Formal Concept Analysis - Mathematical Foundations*. Springer Verlag, 1999.

A. Gómez-Pérez, O. Corcho, and M. Fernández-López. Ontoweb deliverable 1.1.2: Technical roadmap, 2002.

A. Gómez-Pérez, D. Manzano Macho, E. Alfonseca, R. Nú nez, I. Blascoe, S. Staab, O. Corcho, Y. Ding, J. Paralic, and R. Troncy. Ontoweb deliverable 1.5: A survey of ontology learning methods and techniques, 2003.

C. Gooi and J. Allan. Cross-document coreference on a large-scale corpus. In *Proceedings of the Meeting of the North American Association of Computational Linguistics*, 2004.

R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. In *In Proc. of the 16th Int'l Conf. on Computational Linguistics*, Copenhagen, 1996.

C. Grozea. Finding optimal parameters settings for high performance word sense disambiguation. In *Proceedings of the Senseval-3: The thrid International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, 2004.

T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2): 199–220, 1993.

K. M. Gupta, D. W. Aha, E. Marsh, and T. Maney. An architecture for engineering sublanguage wordnets. In *Proceedings of the 1st International Conference of General WordNet*, Mysore, India, 2002.

J. A. Guthrie, L. Guthrie, Y. Wilks, and H. Aidinejad. Subject-dependent co-occurrence and word sense disambiguation. In *Proceedings of the 29th Annual Meeting on Association for Computational Linguistics*, pages 146–152, Berkeley, CA, 1991.

U. Hahn and K. Schnattinger. Towards text knowledge engineering. In *AAAI/IAAI*, pages 524–531, 1998.

S. Harabagiu and D. Moldovan. Question answering. In *Ruslan Mitkov (ed.) The Oxford Handbook of Computational Linguistics*, pages 560–582. Oxford University Press, 2003.

S. Harabagiu and D. I. Moldovan. Knowledge processing in an extended wordnet. In *WordNet: An Electronic Lexical Database*, pages 379–405. MIT Press, 1998.

S. Harabagiu, G. Miller, and D. Moldovan. Wordnet 2 - a morphologically and semantically enhanced resource. In *Proc. of the SIGLEX Workshop on Multilingual Lexicons, ACL Annual Meeting*, University of Maryland, College Park, 1999.

Tobias Hawker. Usyd: Wsd and lexical substitution using the web1t corpus. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 446–453, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

M. A. Hearst. Noun homograph disambiguation using local context in large text corpora. In *Proceedings of the 7th Annual Conference of the Centre for the New OED and Text Research: Using Corpora*, pages 1–22, Oxford, UK, 1991.

M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING-92*, Nantes, France, 1992.

M. A. Hearst. Automated discovery of wordnet relations. In *Christiane Fellbaum (Ed.) WordNet: An Electronic Lexical Database*, pages 132–152. MIT Press, 1998.

J. Hendler. Agents and the semantic web. *IEEE Intelligent Systems Journal*, March/April, 2001.

A. Herbelot and A. Copestake. Acquiring Ontological Relationships from Wikipedia Using RMRS. In *ISWC 2006 Workshop on Web Content Mining with Human Language Technologies, Athens, Georgia*, 2006.

D. Hindle. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association of Computational Linguistics*, pages 268–275, Pittsburgh, 1990.

G. Hirst and D. St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In *WordNet: an electronic lexical database*. MIT Press, 1998.

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. Ontonotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, 2006.

R. Huang, L. Sun, and Y. Feng. Study of Kernel-Based Methods for Chinese Relation Extraction. *Lecture Notes in Computer Science*, 4993:598–604, 2008.

C. H. Hwang. Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information. In *Proceedings of the 6th International Workshop on Knowledge Representation and Data Bases (KRDB)*, Linkoping, Sweden, 1999.

N. Ide and J. Véronis. Machine readable dictionaries: What have we learned, where do we go? In *Proceedings of the Post-Coling94 International Workshop on directions of lexical research*, pages 137–146, Beijing, China, 1994.

N. Ide and J. Véronis. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics*, 24:1–40, 1998.

H. Isozaki and H. Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7, Morristown, NJ, 2002.

J. Jiang and C.X. Zhai. A Systematic Exploration of the Feature Space for Relation Extraction. In *Proceedings of NAACL HLT*, pages 113–120, 2007.

J. S. Justeson and S. M. Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27, 1995.

D. Kalashnikov, M. Sharad, Z. Chen, R. Nuray-Turan, and N. Ashish. Disambiguation algorithm for peopple search on the web. In *Proceedings of IEEE International Conference on Data Engineering*, 2007.

N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics Morristown, NJ, USA, 2004.

R. M. Kaplan. Syntax. In *Ruslan Mitkov (ed.) The Oxford Handbook of Computational Linguistics*, pages 70–90. Oxford University Press, 2003.

L. Khan and F. Luo. Ontology construction for information selection. In *Proceedings of the 14th. IEEE International Conference on Tools with Artificial Intelligence*, Washington D.C., 2002.

J. Kietz, A. Maedche, and R. Volz. A method for semi-automatic ontology acquisition from a corporate intranet. In *Workshop "Ontologies and text", co-located with EKAW'2000*, Juan-les-Pins, France, 2000.

A. Kilgarriff. Senseval: An exercise in evaluating word sense disambiguation programs. In *Proceedings of LREC*, pages 581–588, Granada, May 1998.

A. Kilgarriff and M. Palmer. Introduction to the Special Issue on SENSEVAL. *Computers and the Humanities*, 34(1):1–13, 2000.

D. Klein, J. Smarr, H. Nguyen, and C. Manning. Named entity recognition with character-level models. In *Proceedings of the seventh Conference on Natural Language Learning*, pages 180–183, Edmonton, Canada, 2003.

M. Klein and D. Fensel. Ontology versioning on the semantic web. In *Proceedings of the 1st. International Semantic Web Working Symposium (SWWS)*, pages 75–91, Stanford University, California, 2002. IOS press.

Z. Kozareva, O. Ferrandez, and A. Montoyo. Combining data-driven systems for improving named entity recognition. *Natural Language Processing and Information Systems. Lecture Notes in Computer Sciences*, 3513:80–90, 2005.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL-08: HLT*, pages 1048–1056, Columbus, Ohio, June 2008. Association for Computational Linguistics.

A. Kulkarni and T. Pedersen. Name discrimination and email clustering using unsupervised clustering and labeling of similar contexts. In *Proceedings of the second Indian Conference on Aritifical Intelligence*, 2005.

S. Lappin. Semantics. In *Ruslan Mitkov (ed.) The Oxford Handbook of Computational Linguistics*, pages 91–111. Oxford University Press, 2003.

Y. Keok Lee, H. Tou Ng, and T. Kiah Chia. Supervised word sense disambiguation with support vector machines and multiple knowledge sources. In *Proceedings of the Senseval-3: The thrid International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, 2004.

G. Leech and M. Weisser. Pragamatics and dialogue. In *Ruslan Mitkov (ed.) The Oxford Handbook of Computational Linguistics*, pages 136–156. Oxford University Press, 2003.

W. Lehrnert, C. Cardie, D. Fisher, J. McArthy, E. Riloff, and S. Soderland. Evaluating an information extraction system. *Journal of Integrated Computer-Aided Engineering*, 1(6), 1994.

S. Lemmetty. *Review of Speech Synthesis Technology*. Helsinki University of Technology. Laboratory of Acoustics and Audio Signal Processing, 1999.

D. Lenat. *Steps to Sharing Knowledge*. Mars N., editor, Towards Very Large Knowledge Bases. IOS Press, 1995.

D. B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems*. Addison-Wesley, Reading (MA), USA, 1990.

M. Lesk. Automatic sense disambiguation using machine readable dictionaries. In *Proceedings of the 5th International Conference on Systems Documentation*, pages 24–26, 1986.

X. Li, P. Morie, and D. Roth. Semantic integration in text. *Artificial Intelligence Magazine*, 26(1):45–58, 2005a.

Y. Li, K. Bontcheva, and H. Cunningham. Using uneven margins svm and perceptron for information extraction. In *Proceedings of the ninth Conference on Computational Natural Language Learning*, pages 72–79, Ann Arbor, MI, 2005b.

D. Lonsdale, Y. Ding, D. W. Embley, and A. Melby. Peppering knowledge sources with salt; boosting conceptual content for ontology generation. In *Proceedings of the AAAI Workshop on Semantic Web Meets Language Resources*, Edmonton, Canada, 2002.

C. Macleod and R. Grishman. Comlex syntax reference manual, 1994.

A. Maedche and S. Staab. Discovering conceptual relations from text. In *Technical Report 399, Institute AIFB, Karlsruhe University*, 2000.

A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent systems*, 16(2), 2001.

A. Maedche and R. Volz. The ontology extraction maintenance framework Text-To-Onto. In *Proceedings of the Workshop on Integrating Data Mining and Knowledge Management ICDM-01*, 2001.

Bernardo Magnini and Gabriela Cavaglià. Integrating subject field codes into wordnet. In *Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation*, pages 1413–1418, Athens, Greece, 2000.

M. Magnus. *What's in a word? Studies in phonosemantics*. Ph.D. thesis. Norwegian University NTNU, Faculty of Arts. Department of Linguistics, Science and Technology, 2001.

G.S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of CoNLL-2003*, 2003.

G.S. Mann and D. Yarowsky. Multi-field information extraction and cross-document fusion. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL 2005).*, 2005.

C. D. Manning and H. Schütze. *Foundations of statistical Natural Language Processing*. MIT Press, 2001.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

Y. Matsumoto. Lexical knowledge acquisition. In *R. Mitkov (ed.), Oxford Handbook of Computational Linguistics*, pages 395–413. Oxford University Press, 2003.

J. Mayfield, P. McNamee, and C. Piatko. Named entity recognition using hundreds of thousands of features. In *Proceedings of the seventh Conference on Natural Language Learning*, pages 184–187, Edmonton, Canada, 2003.

J. L. McClelland and D. E. Rumelhart. An interactive activation of context effects in letter perception. *Psychological Review*, 88:375–407, 1981.

O. Medelyan, C. Legg, D. Milne, and I.H. Witten. Mining Meaning from Wikipedia. *Arxiv preprint arXiv:0809.4530*, 2008a.

O. Medelyan, I.H. Witten, and D. Milne. Topic Indexing with Wikipedia. In *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI 2008)*, 2008b.

R. Mihalcea. Using Wikipedia for Automatic Word Sense Disambiguation. In *Proceedings of NAACL HLT*, pages 196–203, 2007.

R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM New York, NY, USA, 2007.

R. Mihalcea and D. Moldovan. A method for word sense disambiguation of unrestricted text. In *Proceedings of ACL'99*, Maryland, NY, 1999.

R. Mihalcea, T. Chkloski, and A. Kilgarriff. The senseval-3 english lexical sample task. In *Proceedings of the Senseval-3: The thrid International Workshop on the Evaluation of Systems for the Semantic Analysis of Text.*, Barcelona, Spain, 2004a.

R. Mihalcea, P. Tarau, and E. Figa. Pagerank on semantic networks, with applications to word sense disambiguation. In *Proceedings of the 20st International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland, 2004b.

Rada Mihalcea, Andras Csomai, and Massimiliano Ciaramita. Unt-yahoo: Supersenselearner: Combining senselearner with supersense and other coarse semantic features. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 406–409, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

A. Mikheev. Periods, capitalized words, etc. *Computational Linguistics*, 28(3):245–288, 2002.

A. Mikheev, C. Grover, and M. Moens. Description of the ltg system used for muc-7. In *Proceedings of 7th Message Understanding Conference (MUC-7)*, 1998.

A. Mikheev, M. Moens, and C. Grover. Named entity recognition without gazeteers. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, Bergen, Norway, 1999.

G. A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

G. A. Miller, C. Leacock, T. Randee, and R. Bunker. A semantic concordance. In *Proceedings of the 3 DARPA Workshop on Human Language Technology*, pages 303–308, 1993.

G. A. Miller, M. Chodorow, S. Landes, C.Leacock, and R. G. thomas. Using a semantic concordance for sense identification. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 240–243, Plainsboro, NJ, 1994.

D. Milne and I.H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. ACM New York, NY, USA, 2008.

M. Milosavljevic, R. Dale, S. J. Green, C. Paris, and S. Williams. Virtual museums on the information superhighway: Prospects and potholes. In *Proceedings of CIDOC'98, the Annual Conference of the International Committee for Documentation of the International Council of Museums*, Melbourne, Australia, 1998.

M. Missikoff, R. Navigli, and P. Velardi. The usable ontology: An environment for building and assesing a domain ontology. In *International Semantic Web Conference (ISCW)*, Sardinia, Italy, 2002.

R. Mitkov. *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2003.

S. Mohanty, N. B. Ray, R. C. B. Ray, and P. K. Santi. Oriya wordnet. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, january 2002.

D. I. Moldovan and R. C. Girju. Domain-specific knowledge acquisition and classification using wordnet. In *Florida Artificial Intelligence Research Society conference, FLAIRS-2002*, Pensacola, Florida, 2002.

R. Navigli and P. Velardi. Ontology learning and its application to automated terminology translation. *IEEE Intelligent Systems*, 18(3):323–340, 2003.

R. Navigli and P. Velardi. Learning domain ontologies from document warehouses and dedicated websites. *Computational Linguistics*, 30, 2004.

H. T. Ng and J. Zelle. Corpus-based approaches to semantic interpretation in natural language processing. *AI Magazine*, 18(4):45–64, 1997.

D.P.T. Nguyen, Y. Matsuo, and M. Ishizuka. Exploiting Syntactic and Semantic Information for Relation Extraction from Wikipedia. In *IJCAI Workshop on Text-Mining & Link-Analysis (TextLink 2007)*, 2007a.

D.P.T. Nguyen, Y. Matsuo, and M. Ishizuka. Relation Extraction from Wikipedia Using Subtree Mining. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, page 1414. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007b.

C. Nicolae and G. Nicolae. Bestcut: A graph algorithm for coreference resolution. In *Proceedings of the 2006 conference on Empirical Methods in Natural Language Processing EMNLP-2006*, pages 275–283, 2006.

S. Nirenburg and V. Raskin. Formal ontology and the needs of ontological semantics. In *J. Pustejovsky (ed.), Ontological Semantics*, chapter 5, pages 134–156. The MIT Press, Cambridge, Massachusetts, 2004.

Joakim Nivre and Mario Scholz. Deterministic dependency parsing of english text. In *Proceedings of Coling 2004*, pages 64–70, Geneva, Switzerland, Aug 23–Aug 27 2004. COLING.

J. Nobécourt. A method to build formal ontologies from text. In *Proceedings of the 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW)*, Juan-les-Pins, France, 2000.

A. Novischi. Accurate semantic annotation via pattern matching. In *Florida Artificial Intelligence Research Society conference, FLAIRS*, Pensacola, Florida, 2002.

P. Pantel and M. Pennacchiotti. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *21st. International COnference on Computational Linguistics ad 44th. Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, Sydney, Australia, 2006.

M. Paşca. Organizing and searching the world wide web of facts–step two: harnessing the wisdom of the crowds. In *Proceedings of the 16th international conference on World Wide Web*, pages 101–110. ACM Press New York, NY, USA, 2007.

M. Pasca and B. Van Durme. What you seek is what you get: Extraction of class attributes from query logs. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2832–2837, 2007.

V. Pekar and S. Staab. Word classification based on combined measures of distributional and semantic similarity. In *Proceedings of Research Notes of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary, 2003.

M. Pennacchiotti and P. Pantel. Ontologizing Semantic Relations. In *21st. International COnference on Computational Linguistics ad 44th. Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, Sydney, Australia, 2006.

F. Pereira, N. Tishby, and L. Lee. Distributional clustering of english words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Columbus, OH, 1999.

P. Devi Poongulhali, N. Kavitha Noel, R. Preeda Lakshmi, T. V. Geetha, and A. Manavazhahan. Tamil wordnet. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, january 2002.

B. Popov, A. Kiryakov, D. Manov, A. Kirlov, D. Ognyanov, and M. Goranov. Towards semantic web information extraction. In *Proceedings of the 2nd International Semantic Web Conference. Workshop on Human Language Technology for the Semantic Web and Web Services.*, Sanibel Island, Florida., 2003.

S. Pradhan, E. Loper, D. Dligach, and M. Palmer. SemEval-2007 Task 17: English Lexical Sample, SRL and All Words. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, 2007.

M. R. Quillian. The teachable language comprehender: A simulation program and theory of language. *Communications of the ACM*, 12(8):459–476, 1969.

L. R. Rabiner and B. H. Huang. An introduction to hidden markov models. *IEEE Acoustics Speech and Signal Processing (ASSP Magazine)*, 3(1):4–16, 1986.

A. Ramsay. Discourse. In *Ruslan Mitkov (ed.) The Oxford Handbook of Computational Linguistics*, pages 112–135. Oxford University Press, 2003.

L. Ranshaw and R. Weischedel. Information extraction. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, 2005.

A. Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. Dissertation. University of Pennsylvania, 1998.

D. Ravichandran and E. Hovy. Learning surface text patterns for a a question answering system. In *Proceedings of the Annual Meetign of the Association of Computational Linguistics (ACL 2002)*, pages 41–47, 2002.

P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 448–453, Montreal, Canada, 1995a.

P. K. Resnik. Disambiguating noun groupings with respect to wordnet senses. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 54–68, Somerset, 1995b. ACL.

S.D. Richardson, W. B. Dolan, and L. VanderWende. Mindnet: Acquiring and structuring semantic information from text. In *Proceedings of the 17th International Conference on Computational Linguistics COLING-ACL*, Montreal, Canada, 1998.

G. Rigau. *Automatic Acquisition of Lexical Knowledge from MRDs*. PhD Thesis, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, 1998.

E. Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thristeen National Conference on Artificial Intelligence (AAAI)*, 1996.

C. Roux, D. Proux, F. Rechermann, and L. Julliard. An ontology enrichment method for a pragmatic information extraction system gathering data on genetic interactions. In *Proceedings of the ECAI00 Workshop on Ontology Learning*, Berlin, Germany., 2000.

M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic inference of word meaning using phonosemantic patterns. In *Second International Semantic Web Conference (ISWC 2004)*, Hiroshima, Japan, 2004.

M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic assignment of wikipedia encyclopedic entries to wordnet synsets. In *Proceedings of the 3rd. Atlantic Web Intelligence Conference AWIC 2005*, volume 3528 of *Lecture Notes in Computer Science*, pages 380–386. Springer-Verlag, 2005a.

M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia. In *Proceedings of the 10th Intl. Conference on Application of Natural Language to Information Systems (NLDB 2005)*, pages 67–79, Alicante, Spain., 2005b.

M. Ruiz-Casado, E. Alfonseca, and P. Castells. Using context-window overlapping in synonym discovery and ontology extension. In *Proceedings of the International Workshop in Text Mining Research, Practice and Opportunities, in conjunction with RANLP conference*, Borovets, Bulgary., 2005c.

M. Ruiz-Casado, E. Alfonseca, and P. Castells. From wikipedia to semantic relationships: a semi-automated annotation approach. In *Proceedings of the first Workshop on Semantic Wikis: From Wiki to Semantics. At the 3rd. European Semantic Web Conference (ESWC 2006)*, Montenegro, 2006.

M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatising the learning of lexical patterns: an application to the enrichment of wordnet by extracting semantic relationships from wikipedia. *Journal of Data and Knowledge Engineering*, 61(3):484–489, 2007.

M. Ruiz-Casado, E. Alfonseca, and P. Castells. *Automatic Acquisition of Semantics from Text for Semantic Work Environments. In J. Rech and B. Decker and E. Ras (Ed.) Emerging Technologies for Semantic Work Environments*, pages 217–244. Information Science Reference, London, 2008a.

M. Ruiz-Casado, E. Alfonseca, M. Okumura, and P. Castells. *Information Extraction and Semantic Annotation of Wikipedia In P. Buitelaar and P. Cimiano (Ed.) Ontology Learning and Population: bridging the gap between Text and Knowledge*, pages 145–170. IOS Press, Amsterdam, 2008b.

V. Rus. *Logic Form For WordNet Glosses and Application to Question Answering*. Ph.D. thesis. Computer Science Department, Southern Methodist University, 2002.

S. Sekine. On-demand information extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association of Computationl Linguistics*, Sydney, Australia, 2006.

S. L. Small and C. Rieger. Parsing and comprehending with word experts (a theory and its realization). In *Wendy Lenhert and Martin Ringle (ed.) Strategies for Natural Language Processing*, pages 89–147. Lawrence Erlbaum and Associates, Hilldale, NJ, 1982.

R. Snow, D. Jurafsky, and A.Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 801–808. Association for Computational Linguistics Morristown, NJ, USA, 2006.

S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1–3):233–272, 1999.

M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

M. Stevenson and Y. Wilks. Word-sense disambiguation. In *Ruslan Mitkov (ed.) The Oxford Handbook of Computational Linguistics*, pages 249–265. Oxford University Press, 2003.

C. Strappavara, A. Gliozzo, and C. Giuliano. Pattern abstraction and term similarity for word sense disambiguation:irst at senseval-3. In *Proceedings of the Senseval-3: The thrid International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, 2004.

F.M. Suchanek, G. Ifrim, and G. Weikum. LEILA: Learning to Extract Information by Linguistic Analysis. In *Proceedings of the ACL-06 Workshop on Ontology Learning and Population*, 2006.

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

M. Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the Second International Conference on Information and Knowledge Base Management, CKIM*, pages 67–74, Arlington, VA, 1993.

P.P. Talukdar, J. Reisinger, M. Pasca, D. Ravichandran, R. Bhagat, and F. Pereira. Weakly-Supervised Acquisition of Labeled Class Instances using Graph Random Walks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 581–589, 2008.

K. Tokunaga, J. Kazama, and K. Torisawa. Automatic discovery of attribute words from web documents. In *Proceedings of IJCAI-2005*, pages 106–118. Springer, 2005.

Stephen Tratz, Antonio Sanfilippo, Michelle Gregory, Alan Chappell, Christian Posse, and Paul Whitney. Pnnl: A supervised maximum entropy approach to word sense disambiguation. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 264–267, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

H. Trost. Morphology. In *Ruslan Mitkov (ed.) The Oxford Handbook of Computational Linguistics*, pages 25–47. Oxford University Press, 2003.

D. Tufis, D. Cristea, and S. Stamou. BalkaNet: Aims, methods, results and perspectives. a general overview. *Romanian Journal on Information Science and Technology. Special Issue on BalkaNet*, 7(1-2):9–34, 2004.

B. Van Durme and M. Pasca. Finding Cars, Goddesses and Enzymes: Parametrizable Acquisition of Labeled Instances for Open-Domain Information Extraction. In *Twenty-Third AAAI Conference on Artificial Intelligence*, 2008.

M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic Wikipedia. In *Proceedings of the 15th international conference on World Wide Web*, pages 585–594. ACM New York, NY, USA, 2006.

E. M. Vorhees. Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 171–180, Pittsburg, PA, 1993.

P. Vossen. *EuroWordNet - A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, 1998.

A. Wagner. Enriching a lexical semantic net with selectional preferences by means of statistical corpus analysis. In *Proceedings of the ECAI00 Workshop on Ontology Learning*, Berlin, Germany., 2000.

R.A. Wagner and M.J. Fischer. The string-to-string correction problem. *Journal of Assoc. Comput. Mach.*, 21, 1974.

G. Wang, H. Zhang, H. Wang, and Y. Yu. Enhancing Relation Extraction by Eliciting Selectional Constraint Features from Wikipedia. *LECTURE NOTES IN COMPUTER SCIENCE*, 4592:329, 2007.

D. Weld, R. Hoffmann, and F. Wu. Using Wikipedia to bootstrap open Information Extraction. *ACM SIGMOD*, 37(4):62–68, 2008.

Y. Wilks. Right attachment and preference semantics. In *Proceedings of the second conference on European chapter of the Association for Computational Linguistics*, pages 89–92, Morristown, NJ, USA, 1985. Association for Computational Linguistics.

Y. Wilks, D. C. Fass, C. Ming Guo, J. E. McDonald, T. Plate, and B. M. Slator. Providing machine tractable dictionary tools. In *James Pustejovsky (ed.) Semantics and the Lexicon*, pages 341–401. Kluwer Academics Publishers, Cambridge, MA, 1990.

F. Xu, D. Kurz, J. Piskorski, and S. Schmeier. A domain adaptive approach to automatic acquisition of domain relevant terms and their relations with bootstrapping. In *Language Resources and Evaluation (LREC-2002)*, Las Palmas, 2002.

D. Yarowsky. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proceedings of the 14th International Conference on Computational Linguistics COLING*, pages 454–460, Nantes, France, 1992.

D. Yarowsky. One sense per collocation. In *Proceedings of ARPA Human Language Technology Workshop*, pages 266–271, Princeton, NJ, 1993.

H. Zaragoza, J. Atserias, M. Ciaramita, and G. Attardi. Semantically annotated snapshot of the english wikipedia v.1 (sw1). `http://www.yr-bcn.es/semanticWikipedia`, 2007.

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

S. Zhao and R. Grishman. Extracting Relations with Integrated Information Using Kernel Methods. *Ann Arbor*, 100, 2005.