Natural language processing for documentation analysis

ITC-irst

Abstract. In view of the increasing interest in ontologies as a source of world knowledge, this deliverable presents different types of ontologies and describes the approach adopted within the Klase project towards the problem of mapping specialized linguistic ontologies to generic resources. It reports on investigations related to the possibility of applying linguistic ontologies to the problem of interoperability of taxonomic structures and to text summarization.

Document Identifier	Deliverable D5.1
Project	MIUR-FIRB project RBNE0195K5 "Knowledge Level Automated Software Engineering"
Version	v1.0
Date	October 13, 2006
State	Final
Distribution	Public

Acknowledgements.

This document is part of a research project funded by the FIRB 2001 Programme of the “Ministero dell’Istruzione, dell’Università e della Ricerca” as project number RBNE0195K5.

The partners in this project are: Istituto Trentino di Cultura (Coordinator), Università degli Studi di Trento, Università degli Studi di Genova, Università degli Studi di Roma “La Sapienza”, DeltaDator S.p.A..

Executive Summary

The KLASE project addresses the problem of providing methodologies and techniques to support the development of software in the goal/actors paradigms, supporting in particular the acquisition and modeling of requirements, which play a fundamental role in the goal/actors paradigm. We investigate the possibility of employing technologies developed in the field of natural language processing to extract, organize and process information written in textual form that is useful for requirements acquisition.

As there has been an increasing interest in ontologies as a source of world knowledge for many NLP applications, we start by presenting different types of ontologies. Linguistic ontologies are large scale lexical resources with an ontological structure, although with a lesser degree of formalization with respect to formal ontologies. A particular kind of linguistic ontologies is represented by specialized linguistic ontologies, i.e. linguistic ontologies with domain specific coverage, as opposed to ontologies which contain generic knowledge. The importance of specialized ontologies, especially for practical applications, is widely recognized. Their use, however, arises the problem of their mapping to generic resources. This deliverable describes the work done in this direction within KLASE. We implemented a methodology to "plug" specialized linguistic ontologies into global ontologies, based on plug relations connecting concepts in the two ontologies.

In the framework of the KLASE project, we have investigated the possibility of applying linguistic ontologies to the problem of the interoperability of taxonomic structures and to the task of text summarization. As far as taxonomic structures are concerned, we focused on Classification Hierarchies (CHs), which are used to organize large amounts of documents. Unlike previous approaches to interoperability, our approach does not consider the content of the documents classified in the CHs. Rather, the algorithm we have developed takes as input the labels attached to two nodes, interprets them and, exploiting both the knowledge contained in a linguistic ontology and the structure of the CH, returns a mapping relation. As far as text summarization is concerned, we focused on the employment of linguistic ontologies for keyphrase extraction. By providing semantic metadata that characterize a document, keyphrases produce an overview of the content of a document. Keyphrase extraction is a relevant technique for a number of NLP tasks, such as document retrieval and clustering. We have developed LAKE, a system which (i) extracts a list of relevant keyphrases from each document of a cluster, (ii) compares the keyphrase lists for each document and estimates both the relevance and the coverage of each list, and (iii) selects the keyphrase list which maximizes the two parameters and substitutes each keyphrase with the sentence in which it appears, so as to build a summary.

Contents

Chapter 1

Introduction

Natural language processing (NLP) is a subfield of artificial intelligence and linguistics which studies the problems of automated understanding and generation of natural human languages. In particular, natural language understanding systems convert samples of human language into more formal representations that are easier for computer programs to manipulate; on the other hand, natural language generation systems convert information from computer databases into normal-sounding human language.

Natural language recognition seems to require extensive knowledge about the outside world and the ability to manipulate it. In this view, there has been an increasing interest in ontologies for many natural language applications, which has led to the creation of ontologies for different purposes and with different features. Ontologies can be grouped into two main categories, i.e. formal ontologies and linguistic ontologies. Linguistic ontologies (e.g. WordNet) are large scale lexical resources that cover most words of a language, while at the same time also providing an ontological structure where the main emphasis is on the relations between concepts; linguistic ontologies can therefore be seen both as a particular kind of lexical database and as particular kind of ontology. Linguistic ontologies mainly differ from formal ontologies as far as their degree of formalization is concerned. Linguistic ontologies, in fact, do not reflect all the inherent aspects of formal ontologies. As [GMV99] point out, for instance, WordNet's upper level structure shows no distinction between types and roles, whereas most of the original Pangloss [KL94] nodes in the Sensus ontology are actually types; to give a further example, WordNet's hierarchical structure lacks information about mutual disjointness between concepts. Moreover, what distinguishes linguistic ontologies from formal ontologies, is their size: linguistic ontologies are very large (WordNet, for instance, has several dozen thousand synsets), while formal ontologies are generally much smaller.

WordNet [Fel98], the best-known linguistic ontology, is an electronic lexical database where each sense of a lemma belongs to a different synset, i.e. a set of synonyms. Synsets are organized hierarchically by means of hypernymy and hyponymy relations. In WordNet other kinds of semantic relations among synsets are defined (e.g. role relation, part-of

relation and cause relation), so as to build a more rich and complex semantic net. WordNet thus offers two distinct services: a lexicon, which describes the various word senses, and an ontology, which describes the semantic relationships among concepts. As a linguistic ontology, WordNet is strongly language-dependent, but as an ontology it could also be adapted to a cross-language environment using the EuroWordNet multilingual database [Vos98] and mapping synsets into the EuroWordNet InterLingual Index, i.e. the index that links monolingual wordnets for all the languages covered by EuroWordNet. There are several examples of monolingual wordnets for many other languages, such as Dutch, Spanish, Italian, German and Basque.

A particular kind of linguistic ontologies is represented by specialized linguistic ontologies, i.e. linguistic ontologies with domain specific coverage, as opposed to global linguistic ontologies, which contain generic knowledge. Focusing on one single domain, specialized linguistic ontologies often provide many sub-hierarchies of highly specialized concepts, whose lexicalizations tend to assume the shape of complex terms (i.e. multiwords); high level knowledge, on the other hand, tends to be simplified and domain oriented.

Many specialized linguistic ontologies have been developed, especially for practical applications, in domains such as art, geography, and medicine, and the importance of specialized linguistic ontologies is recognized in a number of works. The role of terminological resources for Natural Language Processing is addressed, for instance, by [MA00], who point out that high quality specialized resources such as dictionaries and ontologies are necessary for the development of hybrid approaches to automatic term recognition combining linguistic and contextual information with statistical information.

The use of domain terminologies, however, arises the problem of their mapping to a generic resource. The possibility of merging information at different levels of specificity seems to be a crucial requirement at least in the case of large domains where terminologies include both very specific terms and a significant amount of common terms that may be shared with global ontologies. The global-specialized scenario poses some simplifications with respect to the general problem of merging ontologies at the same degree of specificity [Hov98] ; in particular, in the case of conflicting information, it is possible to define a strong precedence criterion according to which terminological information overshadows generic information.

Our work within the KLASE project tried to go a step further in this direction. Assuming the EuroWordNet model, we implemented a methodology to "plug" specialized linguistic ontologies into global ontologies. The formal apparatus to realize this is based on plug relations that connect *basic concepts* of the specialized ontology to corresponding concepts in the generic ontology. We provide experimental data to support our approach, which has been tested on a global and a specialized linguistic ontology for the Italian language.

In the framework of the KLASE project, we have also investigated the possibility of applying linguistic ontologies (i) to the problem of the interoperability of taxonomic

structures and (ii) to the task of text summarization.

As far as taxonomic structures are concerned, we focused on a specific type of taxonomic structures, e.g. Classification Hierarchies (CHs), which are used to organize large amounts of documents.

Unlike previous approaches to interoperability, our approach does not consider the content of the documents classified in the CHs. Documents, in fact, can be of many different types, depending on the characteristics and uses of the hierarchies themselves (in file systems, for instance, documents can be any kind of file, while in the directories of Web portals we have pointers to Web pages and in commercial catalogs we have product cards or service titles). Our approach, on the other hand, is based on a linguistic analysis which allows to interpret the semantics of the labels describing the nodes of the CH.

More specifically, we have develoed CTXMATCH, an algorithm that takes as input the labels attached to two nodes belonging to different CHs, interprets them and, exploiting both the knowledge contained in a linguistic ontology and the structure of the CH, returns the mapping relation existing between the nodes. We have evaluated the overall performance of CTXMATCH and also the performance of the NLP tools employed by CTXMATCH for the semantic interpretation of the labels over real CHs. The results we have obtained represent a useful benchmark, available for future work in this area.

As far as text summarization is concerned, we focused on the employment of linguistic ontologies for keyphrase extraction. Keyphrases provide semantic metadata that characterize documents, producing an overview of the subject matter and contents of a document. Keyphrases extraction is a relevant technique for a number of NLP tasks, such as document retrieval, Web page retrieval, and document clustering. The use of linguistic ontologies allows for a more controlled keyphrase extraction, as the inclusion of a certain phrase in the ontology may help validating a lexically similar keyphrase that has been extracted automatically.

There are two major tasks related to keyphrases: keyphrase assignment and keyphrase extraction (see [Tur99]). In a keyphrase assignment task there is a predefined list of keyphrases (i.e, a *controlled vocabulary* or *controlled index terms*). These keyphrases are treated as classes, and techniques from *text categorization* are used to learn models for assigning a class to a given document. A document is converted to a vector of features and machine learning techniques are used to induce a *mapping* from the feature space to the set of keyphrases (i.e. labels). The features are based on the presence or absence of various words or phrases in the input documents. Usually a document may belong to different classes. In keyphrase extraction (KE), keyphrases are selected from the body of the input document, without a predefined list. When authors assign keyphrases without a controlled vocabulary, typically about 70% to 80% of their keyphrases appear somewhere in the body of their documents [Tur97].

We have developed LAKE, a system which extracts an ordered (according to their position in the document) list of relevant keyphrases from each document of a cluster. Then it compares the keyphrase lists for each document and estimates both the relevance and

the coverage of each list. Finally, the keyphrase list which maximizes the two parameters is selected as the most representative of the cluster and each keyphrase is substituted with the whole sentence in which it appears, until a 250 word summary is built.

Chapter 2

Linguistic Ontologies

Ontologies have become an important topic in research communities across several disciplines in relation to the key challenge of making the Internet and the Web a more friendly and productive place by filling more meaning to the vast and continuously growing amount of data on the net. The surging interest in the discovery and automatic or semi-automatic creation of complex, multi-relational knowledge structures, in fact, converges with recent proposals from various communities to build a Semantic Web relying on the use of ontologies as a means for the annotation of Web resources.

There is also an increasing interest in linguistic ontologies, such as WordNet, for a variety of content-based tasks, such as conceptual indexing and semantic query expansion to improve retrieval performance. More recently, the role of linguistic ontologies is also emerging in the context of distributed agents technologies, where the problem of meaning negotiation is crucial. A relevant perspective in this direction is represented by linguistic ontologies with domain specific coverage, whose role has been recognized as one of the major topics in many application areas.

Our work tries to go a step further in the direction of the interoperability of specialized linguistic ontologies, by addressing the problem of their integration with global linguistic ontologies. The possibility of merging information at different levels of specificity seems to be a crucial requirement at least in the case of large domains where terminologies include both very specific terms and a significant amount of common terms that may be shared with global ontologies.

The global-specialized scenario poses some simplifications with respect to the general problem of merging ontologies at the same degree of specificity [Hov98] ; in particular, in the case of conflicting information, it is possible to define a strong precedence criterion according to which terminological information overshadows generic information. We assume the EuroWordNet model and propose a methodology to "plug" specialized linguistic ontologies into global ontologies. The formal apparatus to realize this is based on plug relations that connect *basic concepts* of the specialized ontology to corresponding concepts in the generic ontology. We provide experimental data to support our approach, which

has been tested on a global and a specialized linguistic ontology for the Italian language.

The chapter is structured as follows. Section 2.1 presents the main features and uses of linguistic ontologies as opposed to formal ontologies. Section 2.2 describes specialized linguistic ontologies (i.e. with domain specific coverage) as opposed to global linguistic ontologies. Section 2.3 focuses on the problem of their interoperability, and describing the relations and the procedure enabling an integrated access of pairs of global and specialized linguistic ontologies.

2.1 Linguistic ontologies versus formal ontologies

In the recent years the increasing interest in ontologies for many natural language applications has led to the creation of ontologies for different purposes and with different features; therefore, it is worth pointing out the distinction between two main kinds of existing ontologies, i.e. formal and linguistic ontologies.

Linguistic ontologies are large scale lexical resources that cover most words of a language, while at the same time also providing an ontological structure where the main emphasis is on the relations between concepts; linguistic ontologies can therefore be seen both as a particular kind of lexical database and as particular kind of ontology.

Linguistic ontologies mainly differ from formal ontologies as far as their degree of formalization is concerned. Linguistic ontologies, in fact, do not reflect all the inherent aspects of formal ontologies. As [GMV99] point out, for instance, WordNet's upper level structure shows no distinction between types and roles, whereas most of the original Pangloss [KL94] nodes in the Sensus ontology are actually types; to give a further example, WordNet's hierarchical structure lacks information about mutual disjointness between concepts.

Moreover, what distinguishes linguistic ontologies from formal ontologies, is their size: linguistic ontologies are very large (WordNet, for instance, has several dozen thousand synsets), while formal ontologies are generally much smaller.

The duality characterizing linguistic ontologies is reflected in their most prominent features. If we consider the linguistic level, they are strongly language-dependent, like electronic dictionaries, glossaries and all other linguistic resources, which focus on the words used in one specific language (in the case of monolingual resources) or in two or more specific language (in the case of bilingual or multilingual resources). On the other hand, if we consider the semantic level, we can observe that concepts denotated by different words in different languages can be shared, as it happens with the concepts in formal ontologies. In fact it is possible, at least for the core Indo-European languages, to identify a common ontological backbone behind the lexical surface of different languages [GMV99].

WordNet [Fel98], the best-known linguistic ontology, is an electronic lexical database

where each sense of a lemma belongs to a different synset, i.e. a set of synonyms. Synsets are organized hierarchically by means of hypernymy and hyponymy relations. In WordNet other kinds of semantic relations among synsets are defined (e.g. role relation, part-of relation and cause relation), so as to build a more rich and complex semantic net. WordNet thus offers two distinct services: a lexicon, which describes the various word senses, and an ontology, which describes the semantic relationships among concepts.

As a linguistic ontology, WordNet is strongly language-dependent, but as an ontology it could also be adapted to a cross-language environment using the EuroWordNet multilingual database [Vos98] and mapping synsets into the EuroWordNet InterLingual Index, i.e. the index that links monolingual wordnets for all the languages covered by EuroWordNet. There are several examples of monolingual wordnets for many other languages, such as Dutch, Spanish, Italian, German and Basque.

A formal ontology based on linguistic motivation is the Generalized Upper Model (GUM) knowledge base [BMF95], an ontology primarily developed for Natural Language Processing applications. An upper model is an abstract linguistically motivated ontology meeting two requirements at the same time: i) a sufficient level of abstraction in the semantic types employed, as to escape the idiosyncrasies of surface realization and ease interfacing with domain knowledge, and ii) a sufficiently close relationship to surface regularities as to permit interfacing with natural language surface components.

Uses of formal ontologies. Recently ontologies have been used in the context of the Semantic Web. Ontologies can be employed to associate meaning with data and documents found on the Internet thus boosting diverse applications of information-retrieval systems. For the retrieval of information from the Web, [LSR96] propose a set of simple HTML Ontology Extensions to manually annotate Web pages with ontology-based knowledge, which performs high precision but is very expensive in terms of time.

OntoSeek [GMV99] is also based on content, but uses ontologies to find user's data in a large classical database of Web pages. [ES99] use an ontology to access sets of distributed XML documents on a conceptual level. Their approach defines the relationship between a given ontology and a document type definition (DTD) for classes of XML document. Thus, they are able to supplement syntactical access to XML documents by conceptual access.

However, as pointed out by [GMV99], the practical adoption of ontologies in information-retrieval systems is limited by their insufficiently broad coverage and their need to be constantly updated; linguistic ontologies encompass both ontological and lexical information thus offering a way to partly overcome these limitations.

Uses of linguistic ontologies. Linguistic ontologies, and WordNet in particular, are proposed for content-based indexing, where semantic information is added to the classic word-based indexing. As an example, *Conceptual Indexing* [Woo97] automatically orga-

nizes words and phrases of a body of material into a conceptual taxonomy that explicitly links each concept to its most specific generalizations. This taxonomic structure is used to organize links between semantically related concepts, and to make connections between terms of a request and related concepts in the index.

[MM00] designed an IR system which performs a combined word-based and sense-based indexing exploiting WordNet. The inputs to IR systems consist of a question/query and a set of documents from which the information has to be retrieved. They add lexical and semantic information to both the query and the documents, during a preprocessing phase in which the input question and the texts are disambiguated. The disambiguation process relies on contextual information, and identifies the meaning of the words using WordNet.

The proble of sense disambiguation in the context of an IR task has been addressed, among the others, also by [GVCC98]. In a preliminary experiment where disambiguation had been done manually, the vector space model for text retrieval gives better results if WordNet synsets are chosen as the indexing space, instead of word forms.

[DJ01] present an approach where linguistic ontologies are used for information retrieval on the Internet. The indexing process is divided into four steps: i) for each page a flat index of terms is built; ii) WordNet is used to generate all candidate concepts which can be labeled with a term of the previous index; iii) each candidate concept of a page is studied to determine its representativeness of this page content; iv) all candidate concepts are filtered via an ontology, selecting the more representative for the content of the page.

More recently, the role of linguistic ontologies is also emerging in the context of distributed agents technologies, where the problem of meaning negotiation is crucial [BS01a].

2.2 Specialized linguistic ontologies versus global linguistic ontologies

A particular kind of linguistic ontologies is represented by specialized linguistic ontologies, i.e. linguistic ontologies with domain specific coverage, as opposed to global linguistic ontologies, which contain generic knowledge. Focusing on one single domain, specialized linguistic ontologies often provide many sub-hierarchies of highly specialized concepts, whose lexicalizations tend to assume the shape of complex terms (i.e. multiwords); high level knowledge, on the other hand, tends to be simplified and domain oriented.

Many specialized linguistic ontologies have been developed, especially for practical applications, in domains such as art (see the Art and Architecture Getty Thesaurus), geography (see the Getty Thesaurus of Geographical Names), medicine [GPS99], etc. and the importance of specialized linguistic ontologies is widely recognized in a number of works.

The role of terminological resources for Natural Language Processing is addressed, for instance, by [MA00], who point out that high quality specialized resources such as dictionaries and ontologies are necessary for the development of hybrid approaches to automatic term recognition combining linguistic and contextual information with statistical information.

[BS02] address the problem of tuning a general linguistic ontology such as WordNet or GermaNet to a specific domain (the medical domain, in the specific case). This involves both selecting the senses that are most appropriate for the domain and adding novel specific terms. Similarly, [TPT+00], describe a method for adapting a general purpose synonym database, like WordNet, to a specific domain (in this case, the aviation domain), adopting an eliminative approach based on the incremental pruning of the original database.

The use of domain terminologies also arises the problem of the (automatic) acquisition of thematic lexica and their mapping to a generic resource [BS01b, Vos01, LMS02]. As far as automatic term extraction is concerned, [BPZ01] investigate whether syntactic context (i.e. structural information on local term context) can be used for determining "termhood" of given term candidates, with the aim of defining a weakly supervised "termhood" model suitably combining endogenous and exogenous syntactic information.

2.3 Merging global and specialized linguistic ontologies

One of the basic problems in the development of techniques for the Semantic Web is the integration of ontologies. Indeed the Web consists of a variety of information sources, and in order to extract information from such sources, their semantic integration is required.

Merging linguistic ontologies introduces issues concerning the amount of data to be managed (in the case of WordNet we have several dozen thousand synsets), which are typically neglected when upper levels are to be merged [SKD01].

Our work tries to go a step further in the direction of the interoperability of linguistic ontologies, by addressing the problem of the integration of global and specialized linguistic ontologies. The possibility of merging information at different levels of specificity seems to be a crucial requirement at least in the case of domains, such as Economics or Law, that includes both very specific terms and a significant amount of common terms that may be shared by the two ontologies. We assume the EuroWordNet model and propose a methodology to "plug" specialized ontologies into global ontologies, i.e. to access them in conjunction through the construction of an integrated ontology.

Correspondences. A global linguistic ontology and a specialized one complement each other. The one contains generic knowledge without domain specific coverage, the other focuses on a specific domain, providing sub-hierarchies of highly specialized concepts.

This scenario allows some significant simplifications when compared to the general problem of merging two ontologies. On the one hand, we have a specialized ontology, whose content is supposed to be more accurate and precise as far as specialized information is concerned; on the other hand, we can assume that the global ontology guarantees a more uniform coverage as far as high level concepts are concerned. These two assumptions provide us with a powerful precedence criterion for managing both information overlapping and inheritance in the integration procedure.

In spite of the differences existing between the two ontologies, in fact, it is often possible to find a certain degree of correspondence between them. In particular, we have information *overlapping* when the same concept belongs to the global and to the specialized ontology, and *over-differentiation* when a terminological concept has two or more corresponding concepts in the global ontology or the other way round. Finally, some specific concepts referring to technical notions may have no corresponding concept in the global ontology, which means there is a *conceptual gap*; in such cases a correspondence to the global ontology can be found through a more generic concept.

The sections highlighted in the global and the specialized ontology represented in Figure 2.1 reflect the correspondences we typically find between the two kinds on ontologies.

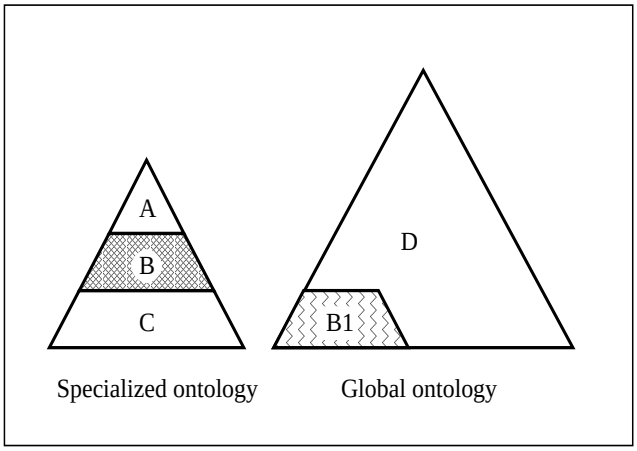

Figure 2.1: Separate specialized and global ontologies. Overlapping is represented in colored areas

As for the global ontology (the bigger triangle), area *B1* is highlighted since it corresponds to the sub-hierarchies containing the concepts belonging to the same specific domain of the specialized ontology (the smaller triangle). The middle part of the specialized ontology, which we call *B* area, is also highlighted and it corresponds to concepts which are representative of the specific domain but are also present in the global ontology.

When the two ontologies undergo the integration procedure, an integrated ontology is constructed (Figure 2.2). Intuitively, we can think of it as if the specialized ontology somehow shifts over the global. In the integrated ontology, the information of the generic is maintained, with the exclusion of the sub-hierarchies containing the concepts belong-

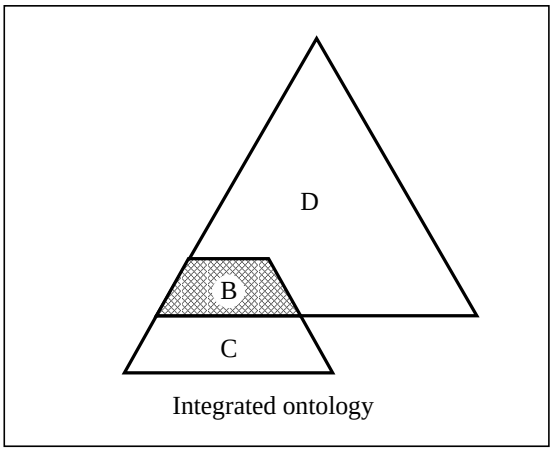

Figure 2.2: Integrated ontology. As to overlapping, precedence is given to the specialized ontology

ing to the domain of the specialized ontology, which are covered by the corresponding area of the specialized. The integrated ontology also contains the most specific concepts of the specialized ontology (*C* area), which are not provided in the generic. What is excluded from the integrated ontology is the highest part of the hierarchy of the specialized ontology; it is represented by area *A* and contains generic concepts not belonging to a specialized domain, which are expected to be treated more precisely in the generic ontology.

Plug relations. The formal apparatus to realize an integrated ontology is based on the use of three different kinds of relations (plug-synonymy, plug-near-synonymy and plug-hyponymy) that connect basic concepts of the specialized ontology to the corresponding concepts in the global ontology, and on the use of eclipsing procedures that shadow certain concepts, either to avoid inconsistencies, or as a secondary effect of a plug relation.

A plug relation directly connects pairs of corresponding concepts, one belonging to the global ontology and the other to the specialized ontology. The main effect of a plug relation is the creation of one or more "plug concepts", which substitute the connected concepts, i.e. those directly involved in the relation. To describe the relations inherited by a plug concept, the following classification, adapted from [HSO98] is used: *up-links* of a concept are those whose target concept is more general (i.e. hypernymy and instance-of relations), *down-links* are those whose target is more specific (i.e. hyponymy and has-instance relations) and *horizontal-links* include all other relations (i.e. part-of relations, cause relations, derivation, etc.).

Plug-synonymy is used when overlapping concepts are found in the global ontology (hereafter *GO*) and in the specialized ontology (hereafter *SO*). The main effect of establishing a relation of plug-synonymy between concept *C* belonging to the global ontology (indicated as C^{GO}) and $C1^{SO}$ (i.e. concept *C1* belonging to the specialized ontology) is

the creation of a plug concept $C1^{PLUG}$. The plug concept gets its linguistic forms (i.e. synonyms) from *SO*, up-links from GO, down-links from *SO* and horizontal-links from *SO* (see Table 1). As a secondary effect, the up relations of $C1^{SO}$ and the down relations of C^{GO} are eclipsed.

	$\boldsymbol{C1}^{PLUG}$
Up links	*GO*
Down links	*SO*
Horizontal links	*GO* + *SO*

Table 2.1: Merging rules for plug-synonymy and plug-near-synonymy.

Plug-near-synonymy is used in two cases: (i) over-differentiation of the *GO*, i.e. when a concept in the *SO* has two or more corresponding concepts in the *GO*; this happens, for instance, when regular polysemy is represented in the GO but not in the SO; (ii) over-differentiation of the *SO*, i.e. when a concept in the *GO* corresponds to two or more concepts in the *SO*; this situation may happen as a consequence of subtle conceptual distinctions made by domain experts, which are not reported in the global ontology. Establishing a plug-near-synonymy relation has the same effect of creating a plug-synonymy (see Table 1).

Plug-hyponymy is used to connect concepts of the specialized ontology to more generic concepts in the case of conceptual gaps. The main effect of establishing a plug-hyponymy relation between C^{GO} (i.e. concept *C* of the global ontology) and $C1^{SO}$ (i.e. concept *C* of the specialized ontology) is the creation of the two plug concepts C^{PLUG} and $C1^{PLUG}$ (see Table 2). C^{PLUG} gets its linguistic forms from the *GO*, up-links from the *GO*, down-links are the hyponyms of C^{GO} plus the link to $C1^{PLUG}$ and horizontal-links from the *GO*. The other plug node, $C1^{PLUG}$, gets its linguistic form from the *SO*, C^{PLUG} as hypernym, down links from the *SO* and horizontal links from the *SO*. As a secondary effect, the hypernym of $C1^{SO}$ is eclipsed.

	$\boldsymbol{C}^{PLUG}$	$\boldsymbol{C1}^{PLUG}$
Up links	*GO*	C^{PLUG}
Down links	*GO* + $C1^{PLUG}$	*SO*
Horizontal links	*GO*	*SO*

Table 2.2: Merging rules for plug-hyponymy

Eclipsing is a secondary effect of establishing a plug relation and is also an independent procedure used to avoid the case that pairs of overlapping concepts placed incon-

sistently in the taxonomies are included in the merged ontology; this could happen, for instance, when ”whale” is placed under a ”fish” sub-hierarchy in a common sense ontology, while also appearing in the mammal taxonomy of a scientific ontology.

Integration procedure. The plug-in approach described in the previous subsection has been realized by means of a semi-automatic procedure with the following four main steps.

(1) Basic concepts identification. The domain expert identifies a preliminary set of ”basic concepts” in the specialized ontology. These concepts are highly representative of the domain and are also typically present in the global ontology. In addition, it is required that basic concepts are disjoint among each other and that they assure a complete coverage of the specialized ontology, i.e. it is required that all terminal nodes have at least one basic concept in their ancestor list.

(2) Alignment. This step consists in aligning each basic concept with the more similar concept of the global ontology, on the basis of the linguistic form of the concepts. Then, for each pair a plug-in configuration is selected among those described in Section 2.3

(3) Merging. For each plug-in configuration an integration algorithm reconstructs the corresponding portion of the integrated ontology. If the integration algorithm detects no inconsistencies, the next plug-in configuration is considered, otherwise step 4 is called.

(4) Resolution of inconsistencies. An inconsistency occurs when the implementation of a plug-in configuration is in contrast with an already realized plug-in. In this case the domain expert has to decide which configuration has the priority and consequently modify the other configuration, which will be passed again to step 2 of the procedure.

Experiments. The integration procedure described in Section 4.3 has been tested within the SI-TAL project [1] to connect a global wordnet and a specialized wordnet that have been created independently. ItalWordNet (IWN) [RAB+00], which was created as part of the EuroWordNet project [Vos98] and further developed through the introduction of adjectives and adverbs, is the lexical database involved in the plug-in as a generic resource and consists of about 45,000 lemmas. Economic-WordNet (ECOWN) is a specialized wordnet for the economic domain and consists of about 5,000 lemmas distributed in about 4,700 synsets. Table 3 summarizes the quantitative data of the two resources considered.

As a first step, about 250 basic synsets (5.3% of the resource) of the specialized wordnet were manually identified by a domain expert, including, for instance “azione” (“share”), and excluding less informative synsets, such as “azione” (“action”). Alignment with respect to the generic wordnet (step 2 of the procedure) is carried out with an algorithm that considers the match of the variants. Candidates are then checked by the domain expert, who also chooses the proper plug relation. In the case of gaps, a synset with a more generic meaning was selected and a plug-hyponymy relation was chosen.

[1] Si-TAL (Integrated System for the Automatic Treatment of Language) is a National Project devoted to the creation of large linguistic resources and software for Italian written and spoken language processing.

	Specialized	Generic
Synsets	4,687	49,108
Senses	5,313	64,251
Lemmas	5,130	45,006
Internal Relations	9,372	126,326
Variants/synsets	1.13	1.30
Senses/lemmas	1.03	1.42

Table 2.3: IWN and ECOWN quantitative data

At this point the merging algorithm takes each plug relation and reconstructs a portion of the integrated wordnet. In total, 4,662 ECOWN synsets were connected to IWN: 577 synsets (corresponding to area *B* in Figure 2.2) substitute the synsets provided in the global ontology to represent the corresponding concepts (*B1* area in Figure 2.1); 4085 synsets, corresponding to the most specific concepts of the domain (*C* area in Figure 2.2) are properly added to the database. 25 high level ECOWN synsets (*A* area in Figure 2.1) were eclipsed as the effect of plug relations. The number of plug relations established is 269 (92 plug-synonymy, 36 plug-near-synonymy and 141 plug-hyponymy relations), while 449 IWN synsets with an economic meaning were eclipsed, either as a consequence of plug relations (when the two taxonomic structures are consistent) or through the independent procedure of eclipsing (when the taxonomies are inconsistent). Each relation connects on average 17,3 synsets.

Chapter 3

Employment of Linguistic Ontologies in Klase: Classification Hierarchies

Classification Hierarchies (CHs) are taxonomic structures used to organize large amounts of documents. Documents can be of many different types, depending on the characteristics and uses of the hierarchy itself. In file systems, documents can be any kind of file; in the directories of Web portals, documents are pointers to Web pages; in the marketplace, catalogs organize either product cards or service titles.

CHs are now widespread as knowledge repositories and the problem of their integration is acquiring a high relevance from a scientific and commercial perspective. In this paper we present CTXMATCH, an algorithm that takes as input the labels attached to two nodes belonging to different partially overlapping CHs and returns a mapping relation (i.e. equivalence, more general, more specific) between them. Unlike previous approaches to interoperability, CTXMATCH does not consider the content of the documents classified in the CHs; rather, it relies both on the semantic interpretation of the labels describing the nodes, which is obtained through a linguistic analysis, and on the structure of the CH itself. The contribution of the paper is in two main directions: (i) we address the linguistic processing required for the semantic interpretation of Ch labels; in our knowledge there are no previous attempts that systematically apply NLP tools and resources to this task; (ii) we report on a large-scale evaluation of the performance of such tools over real CHs. The results we obtained are a useful benchmark, available for future work in this area.

In the attempt to carry out a semantic interpretation over CH nodes, at least the following issues seem to be crucial (examples are taken from Figure 3.1, in which a small subsection of Google Web Directories is reported):

Splitting and contextual interpretation. Information is split on several levels; a single node provides only partial information, so that the interpretation process has to consider a larger scope. As an example, *Players* in Figure 3.1 refers to billiard players, not to players in general.

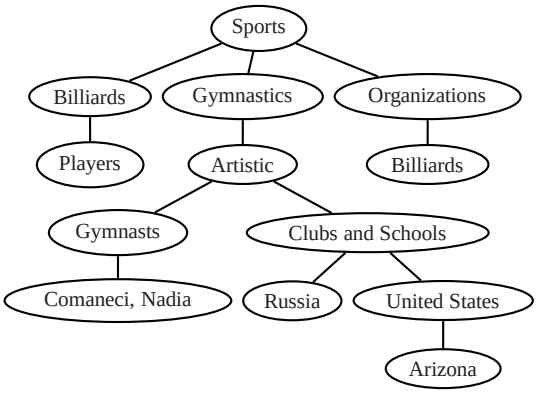

Figure 3.1: Example of a classification hierarchy (from Google Web Directories).

Redundancy. Information can be partially repeated at different levels of a CH. For instance, if *ACL-02* is placed under *Papers-2002*, the fact that *ACL-02* refers to a conference of the year 2002 is implicitly represented at two levels.

Linguistic complexity of the labels. Labels can be arbitrarily complex: they may include abbreviations, multiwords (e.g. *United States* in Figure 3.1), coordinated expressions, proper names (e.g. *Comaneci*, *Nadia*, etc.

Ambiguity and Synonymy. Labels may have different meanings and need to be disambiguated within their context. On the other hand, different labels may have the same meaning (e.g. *Papers* and *Articles*). In order to deal with these aspects of language we have used WORDNET as a repository of senses, and we have designed word sense disambiguation techniques specifically tuned for CHs.

Lack of linguistic context. The interpretation of a label is necessarily based on a limited linguistic context. As a consequence, the application of NLP techniques (e.g. PoS-tagging, word sense disambiguation, etc.) opens up the problem related to the use of tools usually developed for texts,. For instance, we re-trained the PoS-tagger on a specific CH corpus.

Relation to world knowledge. CHs implicitly reflect the world knowledge of a specific domain, but they also reflect the subjective criteria adopted for organizing documents. World knowledge and subjective criteria may interact in subtle ways.

The chapter is structured as follows. In Section 3.1 we review the relevant approaches to interoperability among CHs, outlining the main differences with respect to the semantic-based approach we propose. In Section 3.2 we describe the CTXMATCH algorithm. In Section 3.3 we report on the results of the evaluation experiments where CTXMATCH is applied to the Web Directories of Yahoo! and Google.

3.1 Interoperability of Classification Hierarchies

In our view, the problem of the interoperability among different CHs can be roughly stated in this way: given a node N_s in a source CH and a node N_t in a target CH, the algorithm has to discover a relation between N_s and N_t. Although there can be differences in the definition of the task itself [AS01, MBDH02], and considering that this is a relatively new challenge, approaches to CH mapping can be grouped into four classes, according to the kind of information used: (i) approaches which consider the content of the documents belonging to the CH; (ii) approaches based on the classification of the documents; (iii) approaches that exploit the structure of the CH; and (iv) approaches that attempt a semantic interpretation of the CH labels. In the rest of this Section we will briefly review the first three approaches, while the semantic-based approach will be introduced in more detail in Section 3.2

Mapping based on document content. These approaches rely on the content of the documents classified in a CH. As an example, the GLUE system [DMDH02] employs machine learning techniques to discover mappings among CHs. The idea consists of training a classifier using documents of the source CH, and then apply that classifier to documents of the target CH, and vice-versa. The major drawback of this approach is that it requires textual documents, which prevents its usability when such documents are of a different nature (e.g. images) or they are not available at all.

Mapping based on document classifications. An improvement with respect to the content-based approach has been proposed by Ichise et al. [ITH01], who address the mapping problem by computing a statistical model of the classification criteria of the CHs. Such a statistical model attempts to determine the degree of similarity between two categorization criteria considering the number of documents in common to nodes of different CHs. The advantage over the content-based approach is that the analysis of the documents is not necessary. However, it is required that the source and the target CHs share a certain amount of documents, which is hard to obtain in most of the concrete application scenarios.

Mapping based on structural information. These approaches attempt to discover mappings independently of the number and the type of the classified documents. For instance, Daude et al. [DPR00] exploit a constraint satisfaction algorithm (i.e. relaxation labeling) for discovering relations among ontologies. It first selects candidate pairs using lexical similarities (i.e. concepts with the same label) and then considers a number of structural constraints among nodes (e.g. connections between their hypernyms) to increase or decrease the weights of the connection. Although the approach has been experimented and evaluated to map two versions of WORDNET, achieving high accuracy, our impression is that mapping CHs is a sensibly harder task, due to the highly idiosyncratic way in which CHs may organize their content.

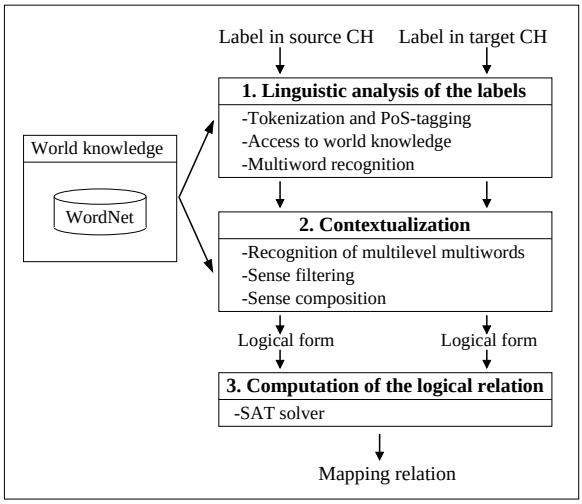

Figure 3.2: The architecture of CTXMATCH.

3.2 CtxMatch: Description

CTXMATCH is a particular implementation of an approach to *semantic coordination* recently proposed in Bouquet et al. [BSZ03] and Magnini et al. [MSS03]. The main difference between CTXMATCH and other approaches to schema matching (see Section 3.1) is that in order to interpret a node of a hierarchy it considers the *implicit information* derived from the *context* where the node occurs, i.e. the structural relations with the other nodes of the hierarchy.

CTXMATCH consists of three main phases: (see Figure 3.2): (i) linguistic analysis of the labels, (ii) contextualization, and (iii) computation of the logical relation.

Linguistic analysis of the labels. In this phase, nodes are interpreted as stand alone objects, i.e. independently of their context and position in the hierarchy.

Words in a label are first tokenized, lemmatized and tagged for Parts of Speech. We use TokenPro and LemmaPro, both developed at IRST, and the TNT tagger [Bra00] with a tag set reduced to the four categories that are significant for accessing WORDNET (i.e. nouns, adjectives, adverbs, verbs), and a generic category 'other'. Then we access a multilingual version of WORDNET developed under the Meaning Project [RMA+02]. When a lemma is found, all the senses provided for the syntactic category selected by the PoS-tagger are attached to the lemma. In the case of *United States* in Figure 3.1, for instance, the WORDNET senses of both the adjective and the noun are added to the label (1).

(1) [`united*`$_a$ `state*`$_n$][1]

When a group of words in a label are contained in WORDNET as a single expression,

[1]We use the following notation: `state*` denotes the disjunction of all the senses of 'state' in WORDNET, while `state#1` indicates sense number 1.

the corresponding senses are selected and the senses of the single lemmas are replaced with those of the multiword. The multiword recognizer we have developed first retrieves the multiwords containing at least two adjacent words of a label and then selects those containing the highest number of words. For instance, 'United States' is recognized as a WORDNET multiword, so this information is added to the label (2).

(2) [$\texttt{United\_States*}_n$]

Then, we transform each label into a formula in description logic [BN02] representing a first approximation of the meaning of a node, where the node is considered a stand alone object.

As a general rule, a label consisting of more than one word is interpreted as the conjunction of all its elements, since the documents classified under a node with a certain label should be concerned with all the words contained in that label; for instance, the label *Laser Games* found in Google Web Directories under under *Sports* is interpreted as [$\texttt{laser*} \sqcap \texttt{game*}$].

Other rules are based on the linguistic material provided in the labels: coordinating conjunctions and commas are interpreted as a disjunction; prepositions are interpreted as a conjunction; expressions denoting exclusion, like 'except', are interpreted as negations. For example, *Clubs and Schools* in Figure 3.1 is interpreted as a disjunction (3), since under that node there might be both documents about clubs and documents about schools.

(3) [$\texttt{club*}_n \sqcup \texttt{school*}_n$]

Contextualization. In the second phase of CTXMATCH we contextualize the interpretation of a node, i.e. we take into consideration its ancestors in order to generate a logical form representing its meaning.

Intuitively, we define the *focus* of a node as the part of the hierarchy that a user is required to visit in order to understand whether a document is under that node. More precisely, given a node N_j in a classification hierarchy *H*, the focus of N_j include all the ancestors of N_j and all their direct descendants in *H*.

The logical form of a node is built combining the logical form of the node with the logical form of its ancestors through intersection. For example, the logical form of the root of the CH in Figure 3.1 is simply [**sport***], while the logical forms of *Billiards* and *Players* contain conjunctions, as shown in 4a and 4b respectively (the label attached to the node to which the logical form refers is highlighted in bold type).

(4a) [$\texttt{sport*}$] $\sqcap$ [**billiards***]
(4b) [$\texttt{sport*} \sqcap \texttt{billiards*} \sqcap$ **player***]

The recognition of multiwords can also be performed on different contiguous levels. For instance, in WORDNET there is a multiword 'billiard player', so in our example (4b), the intersection between `billiards` and `player` is substituted by the senses of the multiword (5).

(5) [$\texttt{sport*} \sqcap \texttt{billiard*\_}$**player***]

The focus of a concept is taken into consideration to perform sense filtering: the senses of N_j that are not compatible given the senses belonging to its focus are deleted. As as example, two senses are attached to *Arizona*, denoting respectively a state in the USA and a snake, and two senses are attached to *United_States*; since there exists a part-of relation between `Arizona#1` and `United_States#1`, and `United_States#1` belongs to the focus of *Arizona*, `Arizona#2` and `United_States#2` can be discarded.

The next step is sense composition, where we address possible inconsistencies between the hierarchical structure and the world knowledge provided in WORDNET. For example, Google Web Directories has *Sociology* and *Science* as sibling nodes under *Academic Study of Soccer*, which admits two conflicting interpretations; from the point of view of the world knowledge provided in WORDNET, `sociology#1` is a second level hyponym of `science#2` (which means that sociology is a science); on the other hand, from the point of view of the hierarchical structure, the sets of documents classified under the two nodes are disjoint). In order to combine the two information sources, *Science* has to be interpreted as if it were *Science except Sociology*.

Computation of the logical relation. We check whether a mapping relation, i.e. an `equivalence`, a `more general` or a `less general` relation, holds between the logical forms *k* and *k'* representing the meaning of the input nodes. To this aim, the task of finding a relation is transformed into a problem of propositional satisfiability (SAT), and then computed via a standard SAT solver. The SAT problem is built in two steps. First, the algorithm selects the portion *T* of the background theory relevant to the two logical forms, namely the semantic relations involving the WORDNET senses that appear in them. Then, it computes some of the logical relations which are implied by *T*. The background theory *T* relevant for computing the relation between two formulas *k* and *k'* is obtained by transforming the WORDNET hierarchical relations between senses appearing in *k* and *k'* into a set of subsumptions in description logic according to the following rules:
- `c#i` $\rightarrow$ `c#j` (if `c#i` is a hyponym of `c#j`);
- `c#j` $\rightarrow$ `c#i` (if `c#i` is a hypernym of `c#j`);
- `c#i` $\equiv$ `c#j` (if `c#i` and `c#j` are synonyms).

The equivalence relation between *k* and *k'* (and thus between the nodes whose meanings are represented by the logical forms) is checked by verifying that $k \sqsubseteq k'$ and $k' \sqsupseteq k$ are both implied by T. Similarly, the `less` [`more`] `general` relation between *k* and *k'* is checked by verifying that $k \sqsubseteq k'$ [$k' \sqsupseteq k$]w is implied by T. For example, the mapping between the source node *Clubs and Schools* in Figure 3.1 and the target node *schools* classified under *athletics/acrobatics/artistic* in a different CHs is one of inclusion. The logical forms of the two nodes (6, 7) and the logical relations implied by the background theory (8, 9) are given to SAT.

(6) [`sport#1`] $\sqcap$ [`gymnastics#1`] $\sqcap$ [`artistic#1`] $\sqcap$ [`club#1` $\sqcup$ `school#1`]
(7) [`athletics#1`] $\sqcap$ [`acrobatics#1`] $\sqcap$ [`artistic#1`] $\sqcap$ [`club#2`]
(8) `sport*` $\equiv$ `athletics#1`
(9) `acrobatics#1` $\rightarrow$ `gymnastics#1`

	Yahoo! Architecture	Google Architecture	Yahoo! Medicine	Google Medicine
# labels	149	413	706	675
# tokens	218	947	1344	931
Tokens/label	1.5	2.3	1.9	1.4
Multiwords/label	0.12	0.09	0.08	0.14
# lexical words	207	700	1137	889
# lemmas not in Wn	7	324	51	30
Wn lemmas coverage	97%	54%	95%	97%
Polysemic lemmas	117	129	672	508
Average polysemy	4	3.7	5.2	3.6

Table 3.1: Analysis of the 'Architecture' and 'Medicine' directories in Yahoo! and Google.

Through SAT we check for satisfiability the union of all the propositions (e.g. 8 and 9) and the negation of the implication between the logical forms 6 and 7. Since the check fails, a `more general` relation is computed between the two nodes; otherwise a similar procedure is followed for the other mapping relations.

3.3 CtxMatch: Evaluation

In this Section we present an experiment performed on the Web Directories of Yahoo! [Yah03] and Google [Goo03] where the outputs of the individual tools and modules we have developed or adapted have been systematically evaluated against a manually tagged gold standard.

The Web Directories of Yahoo! and Google have respectively fourteen and fifteen main categories, each of which can be considered as the root of a CH. For the evaluation of CTXMATCH we have selected the 'Medicine' and the 'Architecture' sub-hierarchies, whose sizes range from one hundred to seven hundred labeled nodes (see Table 3.1). Labels are generally short (on average 1.5-2.3 tokens per label) but nonetheless the occurrence of multiwords is significant (on average, a multiword every ten labels).

WORDNET's coverage with respect to lemmas is generally very high (between 95%and 97% of the lexical words, e.g. nouns, adjectives, verbs and adverbs, are found in WORDNET), with the exception of Google 'Architecture' where it falls to 53.7% (this is due to the fact that more than half the labels consist of names of architects that are not provided in WORDNET). Polysemic lemmas (both single words and multiwords) have on average between 3.6 and 5.2 senses, which makes the need for word sense disambiguation very important.

The evaluation took into consideration (i) tokenization and PoS-tagging; (ii) multiword recognition; (iii) sense filtering: and (iv) logical relation computation. Every phase has been evaluated independently of the errors which occurred in the previous phases, since at every step the algorithm was fed with the correct input built from the gold standard.

Tokenization and PoS-tagging. The performance of the tokenizer was calculated in terms of accuracy with respect to labels: for every label, the output of the tokenizer was evaluated against the gold standard (recall is not significant as the tokenizer always provides an answer). The results (see Table 3.2 show that the performance of the tokenizer is not penalized by the lack of context as, in most cases, we obtained an accuracy of 100%. Only in Google 'Architecture', did the tool make some mistakes (e.g. some middle initials were treated as single letters followed by a full stop).

	Yahoo! Architecture			**Google Architecture**			**Yahoo! Medicine**			**Google Medicine**		
Tokenization (Acc.)	1			.98			1			1		
Lemmatiz. (Acc.)	.97			.98			.99			.98		
PoS-tagging (Acc.)	.96			.90			.97			.90		
Mw. rec. (Pr, Re, F)	.95	1	.97	.95	.95	.95	1	1	1	1	.99	.99
Sense fil. (Pr, Re, F)	.72	.24	.36	.68	.26	.38	.66	.04	.08	.73	.35	.47

Table 3.2: CTXMATCH results on the linguistic analysis of 'Architecture' and 'Medicine' directories for tokenization, lemmatization, PoS-tagging, multiword recognition and sense filtering.

The performance of the lemmatizer and the PoS-tagger are presented in terms of accuracy with respect to single tokens (again, recall is not significant). [2] The evaluation of both tools is not influenced by tokenization errors as the tokens given as input were taken from the gold standard. Accuracy was satisfactory both for lemmatization and PoS-tagging (with rates in the ranges of 97-99% and 90-97% respectively). In most cases, if the selected lemma is wrong, the assigned part of speech is also wrong; however, the cases where the lemma is assigned correctly and the PoS is not (e.g, the plural noun 'States' correctly lemmatized as 'state', but erroneously tagged as verb) are more frequent than the reverse, which explains the slightly better performance in lemmatization.

Multiword Recognition. The performance of the multiword recognizer were more than satisfactory, both in term of precision (correctly retrieved/retrieved) and in terms of recall

[2] Multiple tags were not admitted in the gold standard, so a literal interpretation was preferred; for example, 'New' in 'New York' was tagged as an adjective (only after multiword recognition was it considered as part of the noun multiword `New_York`).

(correctly retrieved/relevant): in total, only three multiwords were missed by the algorithm and three others were misidentified. For example, in the label *Online Databases* in Google 'Medicine', the algorithm did not recognize the multiword `on-line_database` because WORDNET provides only the hyphenated version and the algorithm does not handle this kind of linguistic variation. In *Gropius, Walter* and *Jefferson, Thomas* (in Google 'Architecture'), the algorithm did not recognize `Walter_Gropius` and `Thomas_Jefferson` since it depends on word order (giving up this strict connection to word order would increase recall but would decrease precision).

On the other hand, some false positives occurred because the multiword recognizer does not take into consideration any information about dependency structure and semantics. For example the multiword `city_state` identified by the algorithm in *Traverse City State Hospital* (Google 'Architecture') is wrong in the context of the State Hospital of Traverse City (Michigan) and so are `art movement` in *Arts and Crafts Movement* (Yahoo! 'Architecture') and `Andrew_Jackson` (the US president) in *Downing, Andrew Jackson* (google 'Architecture).

Sense Filtering. The performance of sense filtering is satisfactory as far as precision is concerned: we obtained precision rates varying between 66% and 73%. As an example of wrong sense filtering, in the label *Employment* (placed directly under the root *Medicine*), the algorithm erroneously removes the sense with the meaning of *job* and retains `employment#4` (defined in WORDNET as'the act of using') because of the WORDNET relation between `employment#4` and `optometry#1` (which occurs in the focus of *Employment*).

Since sense filtering strictly depends on the relations found in WORDNET, recall is sensibly lower. In most cases, we obtained satisfactory results, i.e. in the range from 24% to 35%, with a resulting F-measure ranging from 36% to 47%. In the case of Yahoo! 'Medicine', on the other hand, we obtained a recall of 4%. The algorithm actually identified a very low number of WORDNET relations (around hundred) which mainly involved monosemic lemmas (for which no sense filtering is required) and so, in total, sense filtering was applied only to 27 lemmas. This can be explained by the fact that this particular hierarchy contains words which are not much interrelated from the semantic point of view.

Logical Relation Computation. Since it was not feasible to create a manual mapping between all possible pairs of nodes, the logical relations computed by CTXMATCH have been evaluated considering the URLs classified in the CHs. The underlying assumption is that, given a source node and a target node belonging to different hierarchies, the higher the number of the documents (i.e. URLs) shared by the nodes, the higher the similarity between them. The fact that the URLs in Google and Yahoo! Web directories have been classified manually guarantees both that these classifications are of high quality and that they represent a good approximation of human judgment.

The evaluation was performed in four steps: (i) we identified the set *D* of documents classified in both CHs and selected the nodes containing at least one document belonging to this set; (ii) we established a correlation between the proportion of documents shared by source node and target node and the logical relation existing between them. The methodology for this was taken from Doan et al. [DMDH02], who propose three formulas for calculating the similarity between nodes of CHs; (iii) we ran CTXMATCH on the selected nodes; and (iv) evaluated the mapping relations computed by CTXMATCH.

Equivalence relation. The evaluation of the `equivalence` relation is based on the similarity (calculated with the cosine measure) between two sets of documents: the set *A* of documents belonging to the common set of documents *D* classified under the source node, and the set *B* of documents belonging to *D* classified under the target node. According to (10) the similarity between the two sets is 1 if they contain the same documents and 0 if they are disjoint. Since in Yahoo! and Google Web directories the number of documents shared by pairs of nodes is low and there can be different classifications of the same document due to human disagreement, we introduced an approximation factor ϵ, so that an `equivalence` relation is judged as correct if the similarity measure ranges between 1 and (1 - ϵ), where ϵ is empirically set to 0,1.

(10) $SIM(A,B) = A \cap B / \sqrt{(A * B)}$

More [less] general relation. The *most-specific-parent* [*most-general-child*] measure (11) takes a value in the range [0,1] when a node subsumes the other, so a `more [less]` general relation is correct if it ranges between 0 and 1.

$$(11) MSP(A,B) = \begin{cases} P(A|B) & if\ P(B|A) = 1; \\ 0 & otherwise \end{cases}$$

		Prec.	Recall	F-measure
Architecture	equivalence	.33 (.25)	.04 (.04)	.07 (.07)
	more general	.92 (.93)	.42 (.44)	.58 (.60)
	less general	.88 (.90)	.62 (.41)	.73 (.56)
Medicine	equivalence	.27 (.25)	.07 (.05)	.11 (.08)
	more general	.91 (.95)	.48 (.45)	.63 (.61)
	less general	.83 (.86)	.61 (.54)	.70 (.66)

Table 3.3: CTXMATCH (and baseline) results on the mapping of Google and Yahoo! 'Architecture' and Google and Yahoo! 'Medicine'.

The results of the experiment are reported in Table 3.3, in terms of precision, recall, and F-measure obtained for the mapping relations returned by CTXMATCH. A baseline for the experiment was defined by considering a simple string match comparison among the labels placed on the path spanning from a concept to its root in the CH (the results of the baseline are reported in bracket). The results show that both the baseline and the

CTXMATCH algorithm perform quite well. Not surprisingly, the baseline reveals itself as very precise, while CTXMATCH outperforms it with respect to recall. This confirms an important strength of CTXMATCH, namely that a content-based interpretation of contextual knowledge allows the discovery of non-trivial mappings. As an example, the equivalence between the nodes `Pharmacology/Psychopharmacology/Psychiatry` and `Psychiatry/Psychopharmacology` is found thanks to the WORDNET hyponymy relation between `Pharmacology` and `Psychopharmacology`. A mapping of inclusion (source concept is less general than target concept) between `History/Periods_and_Styles/Gothic/Gargoyles` and `History/Medieval` is computed thanks to the relations between `Medieval` and `Gothic` in WORDNET .

Chapter 4

Employment of Linguistic Ontologies in Klase: Text Summarization

Keywords, or keyphrases[1], provide semantic metadata that characterize documents, producing an overview of the subject matter and contents of a document. Keyword extraction is a relevant technique for a number of text-mining related tasks, including document retrieval, Web page retrieval, document clustering and summarization, Human and Machine Readable Indexing and Interactive Query Refinement (see [Tur00] and [GPW+98]).

There are two major tasks exploiting keyphrases: keyphrase assignment and keyphrase extraction (see [Tur99]). In a keyphrase assignment task there is a predefined list of keyphrases (i.e, a *controlled vocabulary* or *controlled index terms*). These keyphrases are treated as classes, and techniques from *text categorization* are used to learn models for assigning a class to a given document. A document is converted to a vector of features and machine learning techniques are used to induce a *mapping* from the feature space to the set of keyphrases (i.e. labels). The features are based on the presence or absence of various words or phrases in the input documents. Usually a document may belong to different classes.

In keyphrase extraction (hereafter KE), keyphrases are selected from the body of the input document, without a predefined list. When authors assign keyphrases without a controlled vocabulary (*free text keywords* or *free index terms*), typically about 70% to 80% of their keyphrases appear somewhere in the body of their documents [Tur97]. This suggests the possibility of using author-assigned free-text keyphrases to train a KE system. In this approach, a document is treated as a set of candidate phrases and the task is to classify each candidate phrases as either a keyphrase or non-keyphrase [Tur97, FPW+99]. A feature vector is calculated for each candidate phrase and machine learning techniques are used to learn a model which classifies each candidate phrase as a keyphrase or non-keyphrase.

[1] Throughout this document we use the latter term to subsume the former.

Our work proposes to exploit a keyphrase extraction methodology in order to identify relevant terms in the document. Afterward, a score mechanism is used to score the best sentences for each cluster of documents. At its heart, the LAKE algorithm first considers a number of linguistic features to extract a list of well motivated candidate keyphrases, then uses a machine learning framework to select significant keyphrases for a document. With respect to other approaches to keyphrase extraction, LAKE makes use of linguistic processors such as named entities recognition, which are not usually exploited.

LAKE participated in the DUC-2004 evaluation exercise, task 1 (*very short single document summaries*, limited to 75 bytes). The system was based on the idea of Keyphrase Extraction as a useful approximation to summarization. We will discuss results and comment on both human assessment (Linguistic Quality and responsiveness of the summaries) and the Pyramid based evaluation.

The chapter is organized as follows. In Section 4.1 we report on the general architecture of our system, which combines a machine learning approach with a linguistic processing of the document. Section 4.2 describes the participation of the system in the DUC-2005 evaluation exercise and Section 4.3 shows the results obtained by the system.

4.1 LAKE

LAKE (Linguistic Analysis based Keyphrase Extractor) is a keyphrase extraction system based on a supervised learning approach which makes use of linguistic processing of documents. The system uses Nave Bayes as the learning algorithm and TF*IDF term weighting with the position of a phrase as features. Unlike other keyphrase exctraction systems, like Kea and Extractor, LAKE chooses the candidate phrases using linguistic knowledge. The candidate phrases generated by LAKE are sequences of Part of Speech containing Multiword expressions and Named Entities. Extraction is driven by a set of ”patterns” which are stored in a pattern database; once there, the main work is done by the learner device. The linguistic database makes LAKE unique in its category.

LAKE is based on three main components, (represented in Figure 1) : the Linguistic Pre-Processor, the candidate Phrase Extractor and the Candidate Phrase Scorer.

Linguistic Pre-Processor. Every document is analyzed by the Linguistic Pre-Processor in the following three consecutive steps: Part of speech analysis, Multiword recognition and Named Entity Recognition.

- Part of Speech Tagger. The Part of Speech (POS) tagger built upon a tokenizer and sentence delimiter, labeling each word in a sentence with its appropriate tag. It decides if a given word is a noun, verb, adjective, etc. The POS tagger adopted by LAKE is the TreeTagger[2], developed at the University of Stuttgart [Sch94]. The

[2]http://www.

TreeTagger uses a decision tree to obtain reliable estimates of transition probabilities. It determines the appropriate size of the context (number of words) which is used to estimate the transition probabilities. For example, if we have to find the probability of a noun appearing after a determiner followed by an adjective we find out whether the previous tag is ADJ; if yes, then we go into the ”yes” branch and check if the tag previous to this was a determiner; if ”yes” then we get to a probability of this occurrence.

- Multiwords Recognition. Sequences of words that are considered as single lexical units are detected in the input document according to their presence in WordNet [Fel98]. For instance, the sequence *Christmas trees* is transformed into the single token *Christmas_tree* and the PoS tag found in WordNet is assigned to it.

- Named Entities Recognition. The task of Named Entity Recognition (NER) requires a program to process a text and identify expressions that refer to people, places, companies, organization, products, and so forth. Thus the program should not merely identify the boundaries of a naming expression, but also classify the expression, e.g., so that one knows that Rome refers to a city and not a person. For Named Entities recognition we used LingPipe[3], a suite of Java tools designed to perform linguistic analysis on natural language data. The tool includes a statistical named-entity detector, a heuristic sentence boundary detector, and a heuristic within-document co reference resolution engine. Named entity extraction models are included for English news and can be trained for other languages and genres.

Candidate Phrase Extractor. Syntactic patterns that described either a precise and well defined entity or concise events/situations were selected as candidate phrases (e.g. phrases that may be selected as document reorientations). In the former case, the focus was on unigrams and bi-grams (for instance Named Entity, noun, and sequences of adjective+noun, etc.), while in the latter have been considered longer sequences of parts of speech, often containing verbal forms (for instance noun+verb+adjective+noun). Sequences such as noun+adjective that are not allowed in English were not taken into consideration. Patterns containing punctuation have been eliminated. Manually have been selected a restricted number of PoS sequences that could have been significant in order to describe the setting, the protagonists and the main events of a newspaper article. To this end, particular emphasis was given to named entities, proper and common names. Once all the uni-grams, bi-grams, tri-grams, and four-grams were extracted from the linguistic pre-processor, they were filtered with the patterns defined above.

As an example, let consider a document belonging to the DUC corpus[4] that reports on the possible extradition of Pinochet from London to Spain. Table refta:duc shows some of the candidate phrases that our largest filter accepted as candidates from this document.

[3]LingPipe is free, available at http://www.alias-i.com/lingpipe/index.html
[4]http://www-nlpir.nist.gov/projects/duc/data.html

Table 4.1: Examples of types of phrases and their patterns

Type of phrase	Pattern	Example
Uni-Gram	NE	London
	NE	1973
Bi-Gram	JJ+NN	Chilean dictator
	JJ+NN	Spanish magistrate
	JJ+NN	urinary infection
Tri-Gram	NN+CC+NN	genocide and terrorism
	NN+VBD+NE	newspaper reported Friday
	NN+VBD+NN	room locked television
Four-Gram	NE+MD+VB+VBN	Augusto Pinochet would be extradited
	VBN+IN+JJ+NNS	detained by British police
	NN+TO+VB+NN	extradition to stand trial
	NN+VBD+JJ+NN	dictatorship caused great suffering

Candidate Phrases Scorer. In this phase a score is assigned to each candidate phrase in order to rank it and allowing the selection of the most appropriate phrases as representative of the original text. The score is based on a combination of TF*IDF (i.e. the product of the frequency of a candidate phrase in a certain document and the inverse frequency of the phrase in all documents) and first occurrence, i.e. the distance of the candidate phrase from the beginning of the document in which it appears. (These features are commonly used keyphrase-related features.) However, since the frequency of a candidate phrase in the whole collection is not significant, candidate phrases do not appear frequently enough in the collection. It has been decided to estimate the values of the TF*IDF using the head of the candidate phrase, instead of the phrase itself. According to the principle of headedness [AvdWKvB00], any phrase has a single word as head. The head is the main verb in the case of verb phrases, and a noun (last noun before any post-modifiers) in noun phrases.

As learning algorithm, it has been used the Naïve Bayes Classifier provided by the WEKA package [WEFF99]. The classifier was trained in the following way on a corpus with the available keyphrases. From the document collection we extracted all the nouns and the verbs. Each of them was marked as a positive example of a relevant keyphrase for a certain document if it was present in the assessor's judgment of that document; otherwise it was marked as a negative example. Then the two features (i.e. TF*IDF and first occurrence) were calculated for each word. The classifier was trained upon this material and a ranked word list was returned (e.g., dictator, magistrate, infection, etc. see Table 1). The system automatically looks in the candidate phrases for those phrases containing these words. In our case Chilean dictator, Spanish magistrate, urinary infection, etc. The top candidate phrases matching the word output of the classifier are

kept. The model obtained is reused in the subsequent steps. When a new document or corpus is ready we use the pre-processor module to prepare the candidate phrases. The model we got in the training is then used to score the phrases obtained. In this case the pre-processing part is the same. So, using the model we got in the training, we extract nouns and verbs from documents, and then we keep the candidate phrases containing them.

4.2 LAKE at DUC-2005

Our decision to participate at DUC-2005 was mainly motivated by the fact that some features of Task 1, i.e. the length limit of the output summaries and the fact that summaries could be returned as lists of disjointed items, seemed to fit well in a KE approach. In futher experiments LAKE has been tested as a useful device in text mining application suitable for small devices as well [DK05]. Still, in [BD02] is discussed the usefulness of KE for knowledge management purposes.

Given a user profile, a DUC topic, and a cluster of documents relevant to the DUC topic, participants were asked to create from the documents a brief, well-organized, fluent summary addressing the need for information expressed in the topic, at the level of granularity specified in the user profile. The summary should not be longer than 250 words (whitespace-delimited tokens) and should include (in some form or other) all the information in the documents that contributes to meeting the information need. Each group was allowed to submit one set of results, i.e., one summary for each topic/cluster. A number of extensions, described in the rest of this Section, were necessary in order to adapt the LAKE system to the new task.

As a first step, we continued to use keyphrases as a document surrogate. In other words, we exploited the LAKE core system abilities to extract from each document j of a cluster an ordered list of keyphrases kl_j. Two options has been added with respect to last year system. First, it is possible to set the number of keyphrases that the system extracts from each document. Second, it is it is possible to set the maximum number of words composing a keypkrase. In short, for a given document j the system is able to extract a keyphrase list kl_j, as long as we like and with the possibility to choose the number of words (i.e. up to four words) contained in each keyphrase of the extracted list.

Then we compare the keyphrase lists for each document and we estimate two measures which we think are crucial for selecting the most representative kl_j among those produced for a certain cluster. both the relevance and the coverage of each list. Given a kl for a document d of a cluster C_j, the next step is to look for a score mechanism able to select the best kl and a as consequence the document that better represents the whole cluster.

A summary for a cluster C is represented by sentences of the document d_j belonging to C_j, which best represents fact reported in C. To estimate the representativeness of a document d in a cluster C we use two measures: the relevance of the document in C and

the coverage of the document in C. Since documents are represented as list of relevant keyphrases, the two measures are computed over such keyphrase list.

The relevance of a keyphrase list kl_j with respect to a cluster C_j is computed considering the frequency of the keyphrases composing the list. The intuition is that keyphrases with higher frequency bring the more relevant information in the cluster. Relevance is calculated according to the following formula:

$$relevance(kl_j) = \frac{\sum_{w=1}^{n} freq(w, kl_j)}{freq(w, C_j)} \tag{4.1}$$

where $freq(w, kl_j)$ is the count of a word w in a certain document and $freq(w, C_j)$ is the count of w in all the document in cluster C_j.

The coverage of a keyphrase list kl_j is an indication of the amount of information that the keyphrase list contain with respect to the total amount of information included in a cluster of documents. Coverage is calculated according to the following formula:

$$coverage(kl_j, C) = \frac{length(kl_j)}{maxlength(kl_j, C)} \tag{4.2}$$

where $length(kl_j)$ is the number of keyphrases extracted from document j whereas $maxlength(kl_j, C)$ is the length of the longest keyphrase list extracted from a document belonging to cluster C_j. The intuition underlying being that the longer the keyphrase list, the more is its coverage for a certain cluster.

Finally, relevance and coverage are combined according to the following formula:

$$rep(kl_j) = relevance(kl_j, C) \times coverage(kl_j, C) \tag{4.3}$$

which gives an overall measure of the representativeness of a keyphrase list for a certain document with respect to a cluster.

Finally, the keyphrase list which maximize the two parameters is selected as the most representative of the cluster and each keyphrase is substituted with the whole sentence in which it appears, until a 250 word summary is built.

4.3 Results

In this Section we discuss results obtained at DUC-2005 and comment on both human assessment (Linguistic Quality and Responsiveness of the summaries) and the Pyramid based evaluation, experimented for the first time this year, for which CELCT (Center for the Evaluation of Language and Communication technologies) has been involved. LAKE

Table 4.2: Results of the LAKE system at DUC 2005

	Average score	**Relative position**
Linguistic Quality	3.968	1/31
Responsiveness (Scaled)	16.7	19/31
ROUGE-2	0.056270211	20/31
ROUGE-SU4	0.1106907611	20/31

scored very well (first position) as far as the Linguistic Quality was concerned, confirming the hypothesis that an ordered list of relevant keywords is a good representation of the document content. As for for the Pyramid evaluation LAKE scored 19 on 31 participants.

Linguistic quality and responviveness. Summaries at DUC-2005 have been evaluated by human assessors according to both their Linguistic Quality and to their Responsiveness. Linguistic quality assesses how readable and fluent the summaries are, and measures the qualities of the summary without comparing it with a model summary or DUC topic. Five *Quality Questions* were used:

1. Grammaticality
2. Non-redundancy
3. Referential clarity
4. Focus
5. Structure and Coherence

All linguistic quality questions were assessed on a five-point scale from ”1” (very poor) to ”5” (very good). As Table 4.2 shows LAKE, in average, obtained very good results in this sense.

As for responsiveness the evaluation assesses how well each summary responds to the topic. After having read the topic statement and all the associated summaries, assessors grade each summary according to how responsive it is to the topic. The score was an integer between 1 and 5, with 1 being least responsive and 5 being most responsive. For a given topic, some summary was required to receive each of the five possible scores, but no distribution was specified for how many summaries had to receive each score. The number of human summaries per topic also varied. Therefore, raw responsiveness scores cannot be directly compared across topics. The result LAKE obtained for *scaled responsiveness* is reported in Table 4.2. As can be seen LAKE scored 19 out of 31 systems participating.

ROUGE based evaluation. A second evaluation was conducted running ROUGE-1.5.5 with the main goal of computing recall scores (i.e., ROUGE-2 and ROUGE-SU4), even though other scores are computed by the system. Table 4.2 reports the results of these two score. For both the evaluations, LAKE scored 20 out of 31 participating systems.

Pyramid based evaluation. ROUGE provides an automatic method to evaluate systems, however, Nenkova et al. (Nenkova and Passoneau, 2004) showed that ROUGE measure cannot be used as an absolute measure of the system's performance. To fill up this gap they proposed the Pyramid approach, that is a manual method for summarization evaluation, developed in an attempt to address the fact that humans choose different words when write a summary. In short, the method seeks to match content units in peer summaries (i.e., produced automatically by the systems) with similar content units found in a pool of human summaries. A good peer summary is one where its contents units are observed across many human summaries.

Table 4.3 and Table 4.4 show the results obtained. LAKE obtained competitive results scoring 11th and 10th, respectively for score (also named original score) and for modified score. The original score uses as X the same number as units appearing in the peer (i.e., it is precision oriented), while the modified score uses as X the average number of units found in the human (model) summaries (i.e., it is recall oriented).

Table 4.3: Results for the Pyramid metric

Peer id	Average Score	Rank Score
14	0.2477	1
17	0.2398	2
10	0.2340	3
15	0.2322	4
7	0.2307	5
4	0.2197	6
16	0.2170	7
32	0.2134	8
6	0.2110	9
19	0.2089	10
12	0.2086	11
11	0.2085	12
21	0.2063	13
26	0.1970	14
28	0.1944	15
3	0.1894	16
13	0.1855	17
25	0.1691	18
1	0.1666	19
27	0.1631	20
31	0.1587	21
24	0.1491	22
20	0.1446	23
30	0.1376	24
23	0.1216	25

Table 4.4: Results for the Pyramid metric

Peer id	Average Modified Score	Rank Score
10	0.2000	1
17	0.1972	2
14	0.1874	3
7	0.1840	4
15	0.1793	5
4	0.1722	6
16	0.1706	7
11	0.1691	8
19	0.1672	9
12	0.1645	10
6	0.1639	11
32	0.1607	12
21	0.1589	13
3	0.1459	14
26	0.1413	15
13	0.1412	16
28	0.1400	17
25	0.1395	18
27	0.1306	19
1	0.1258	20
31	0.1215	21
24	0.1140	22
30	0.1131	23
20	0.0937	24
23	0.0609	25

Bibliography

[AS01] R. Agrawal and R. Srikant. On integrating catalogs. In *Proc. of the Tenth International World Wide Web Conference (WWW-2001)*, Hong Kong, China, May 2001.

[AvdWKvB00] A. Arampatzis, T. van der Weide, C. Koster, and P. van Bommel. An evaluation of linguistically-motivated indexing schemes. In *In Proceedings of the BCSIRSG '2000*, 2000.

[BD02] Luciana Bordoni and Ernesto D'Avanzo. Prospects for integrating text mining and knowledge management. *The IPTS Report*, (68):21–25, 2002.

[BMF95] J.A. Bateman, B. Magnini, and G. Fabris. The generalized upper model knowledge base: Organization and use. In *Proc. of International Conference on Bulding and Sharing of Very Large-Scale Knowledge Bases*, Twente, The Netherlands, April 1995.

[BN02] F. Baader and W. Nutt. Basic description logics. In F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 47–100. Cambridge University Press, 2002.

[BPZ01] R. Basili, M.T. Pazienza, and F.M. Zanzotto. Modelling syntactic context in automatic term extraction. In *Proc. of Recent Advances in Natural Language Processing (RANLP '01)*, Tzigov Chark, Bulgaria, September 2001.

[Bra00] T. Brants. Tnt – A statistical part-of-speech tagger. In *Proc. of the 6th Applied NLP Conference (ANLP-2000)*, Seattle, April-May 2000.

[BS01a] P. Bouquet and L. Serafini. Two formalizations of a context: a comparison. In *Proc. of Third International Conference on Modeling and Using Context*, Dundee, Scotland, July 2001.

[BS01b] P. Buitelaar and B. Sacaleanu. Ranking and selecting synsets by domain relevance. In *Proc. of NAACL Workshop WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburgh, June 2001. held in conjunction with NAACL2001.

[BS02] P. Buitelaar and B. Sacaleanu. Extending synsets with medical terms. In *Proc. of the First Global WordNet Conference*, Mysore, India, January 2002.

[BSZ03] P. Bouquet, L. Serafini, and S. Zanobini. Semantic coordination: A new approach and an application. In *Proc. of the Second International Semantic Web Conference (ISWC 2003)*, Sanibel Island, Florida (USA), October 2003.

[DJ01] E. Desmontils and C. Jacquin. Indexing a web site with a terminology oriented ontology. In *Proc. of SWWS International Semantic Web Working Symposium*, Stanford University, USA, July, August 2001.

[DK05] Ernesto D'Avanzo and Tsvi Kuflik. Linguuistic summaries on small screens. In *Data Mining VI*, pages 195–204. WIT Press, 2005.

[DMDH02] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *Proc. of WWW-2002*, Honolulu, Hawaii, May 2002.

[DPR00] J. Daude, L. Padro, and G. Rigau. Mapping wordnets using structural information. In *Proc. of the 38th Anual Meeting of the Association for Computational Linguistics (ACL-2000)*, Hong Kong, October 2000.

[ES99] M. Erdmann and R. Studer. Ontologies as conceptual models for XML documents. In *Proc. of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management (KAW '99)*, Voyager Inn, Banff, Alberta, Canada, October 1999.

[Fel98] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, US, 1998.

[FPW+99] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. Domain-specific keyphrase extraction. In *IJCAI*, pages 668–673, 1999.

[GMV99] N. Guarino, C. Masolo, and G. Vetere. OntoSeek: Contet-based access to the web. *IEEE Intelligent Systems and Their Application*, 14(3):70–80, 1999.

[Goo03] Google. http://directory.google.com/, 2003.

[GPS99] A. Gangemi, D.M. Pisanelli, and G. Steve. Overview of the ONIONS project: Applying ontologies to the integration of medical terminologies. *Data and Knowledge Engineering*, 31, 1999.

[GPW+98] C. Gutwin, G. Paynter, I. Witten, C. NevillManning, and E. Frank. Improving browsing in digital libraries with keyphrase indexes. Technical report, Department of Computer Science, University of Saskatchewan, Canada, 1998.

[GVCC98] J. Gonzalo, F. Verdejio, Chugur, and J. Cigarran. Indexing with WordNet synsets can improve text retrieval. In S. Harabagiu, editor, *Proceeding of the Workshop "Usage of WordNet in Natural Language Processing Systems"*, Montreal, Quebec, Canada, August 1998.

[Hov98] E. Hovy. Combining and standardizing large-scale, pratical ontologies for machine translation and other uses. In *Proc. of the First International Conference on Language Resources and Evaluation*, Granada, Spain, August 1998.

[HSO98] G. Hirst and D. St-Onge. Lexical chains representations of context for the detection and correction of malapropisms. In C. Fellbaum, editor, *WordNet. An Electronic Lexical Database*. The MIT Press, 1998.

[ITH01] R. Ichise, H. Takeda, and S. Honiden. Integrating multiple internet directories by instance-based learning. In *Proc. of IJCAI-2003*, Acapulco, Mexico, August 2001.

[KL94] K. Knight and S. Luk. Building a large knowledge base for machine translation. In *Proceedings of the American Association of Artificial Intelligence Conference AAAI-94*, Seattle, WA, 1994.

[LMS02] A. Lavelli, B. Magnini, and F. Sebastiani. Building thematic lexical resources by bootstrapping and machine learning. In *Proc. of the Workshop "Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Language Data", Workshop at LREC-2002*, 2002. to appear.

[LSR96] S. Luke, L. Spector, and D. Rager. Ontology-based knowledge discovery on the world-wide-web. In *Proc. of the AAAI1996 Workshop on Internet-based Information Systems*, Portland, Oregon, August 1996.

[MA00] D. Maynard and S. Ananiadou. Creating and using domain-specific ontologies for terminological applications. In *Proc. of Second International Conference on Language Resources and Evaluation (LREC-2000)*, Athens, Greece, May, June 2000.

[MBDH02] J. Madhavan, P. Bernstein, P. Domingos, and A. Halevy. Representing and reasoning about mappings between domain models. In *Proc. of AAAI-2002*, Edmonton, Alberta, July-August 2002.

[MM00] R. Mihalcea and D. Moldovan. Semantic indexing using WordNet senses. In *Proc. of the ACL workshop on Recent Advances in Natural Language Processing and Information Retrieval*, Hong Kong, October 2000.

[MSS03] B. Magnini, L. Serafini, and M. Speranza. Making explicit the semantics hidden in schema models. In *Proc. of the Workshop on 'HLT for the Semantic Web and Web Services', ISWC-2003*, Sanibel Island, Florida, October 2003.

[RAB+00] A. Roventini, A. Alonge, F. Bertagna, B. Magnini, and N. Calzolari. ItalWordNet: a large semantic database for Italian. In *Proc. of the Second International Conference on Language Resources and Evaluation (LREC-2000)*, Athens, Greece, May, June 2000.

[RMA+02] G. Rigau, B. Magnini, E. Agirre, P. Vossen, and J. Carrol. Meaning: A roadmap to knowledge technologies. In *Proc. of the workshop 'A Roadmap for Computational Linguistics', COLING-02*, Taipei, Taiwan, August-September 2002.

[Sch94] Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK, 1994.

[SKD01] K. I. Simov, K. Kiryakov, and M. Dimitrov. OntoMap - the guide to the upper-level. In *Proc. of SWWS International Semantic Web Working Symposium*, Stanford University, USA, July, August 2001.

[TPT+00] D. Turcato, F. Popowich, J. Toole, D. Fass, D. Nicholson, and G. Tisher. Adapting a synonym database to specific domains. In *Proc. of Workshop on Information Retrieval and Natural Language Processing*, Hong-Kong, October 2000. held in conjunction with ACL2000.

[Tur97] P.D. Turney. Extraction of keyphrases from text: Evaluation of four algorithms. Technical Report ERB-1051. (NRC #41550), National Research Council, Institute for Information Technology, 1997.

[Tur99] P.D. Turney. Learning to extract keyphrases from text. Technical Report ERB-1057. (NRC #41622), National Research Council, Institute for Information Technology, 1999.

[Tur00] P.D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2 (4):303–336, 2000.

[Vos98] P. Vossen, editor. *EuroWordNet: a Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, 1998.

[Vos01] P. Vossen. Extending, trimming and fusing WordNet for technical documents. In *Proc. of NAACL Workshop WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburgh, June 2001. held in conjunction with NAACL2001.

[WEFF99] Ian H. Witten, Eibe Frank Eibe Frank Ian H.Witten Eibe Frank, Eibe Frank Ian H. Witten, and EEibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

[Woo97] W.A. Woods. Conceptual indexing: A better way to organize knowledge. Technical report, SUN Technical Report TR-97-61, 1997.

[Yah03] Yahoo. http://uk.yahoo.com/, 2003.