WORD SENSE DISAMBIGUATION AND TEXT RELATEDNESS BASED ON WORD THESAURI

PHD THESIS

DEPARTMENT OF INFORMATICS
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Georgios Tsatsaronis

June 2009


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Preface

As the immense amount of text data increases rapidly over the years, the need to improve the quality of algorithms in text related tasks is eminent. Traditional models for representing documents, like the standard vector space model (VSM), often neglect the semantic relatedness between words, suffering from the restriction of exact keywords matching, in order to explore the similarity or relatedness between segments of text. In critical tasks, like text classification and retrieval, which have been studied over the past decades intensively, this assumption of exact keyword matching is often the reason for poor performance. This thesis aims to explore the potential of incorporating semantic relatedness between documents in several text related applications, like text classification, retrieval and paraphrasing recognition. Several aspects have been taken into account, like natural language processing techniques and use of a word thesaurus, namely WordNet, in an effort to exhaust as many possibilities as possible in the workflow from analyzing and preprocessing documents up to embedding successfully the semantic information in a machine readable manner in those tasks. The outcome of this thesis shows that lexical semantic similarity can be used efficiently in the studied tasks and that it can boost their performance, widening the possibilities of more efficient algorithms in text applications. This thesis is part of the research project number 03EΔ850/8.3.1., implemented within the framework of the Greek Reinforcement Programme of Human Research Manpower (PENED) and co-financed by Greek national and European Union Funds (25% from the Greek Ministry of Development-General Secretariat of Research and Technology, and 75% from E.U.- European Social Fund).

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Acknowledgements

I would like to warmly thank my supervisor Michalis Vazirgiannis, to which I owe great gratitude for his continuous support. Many thanks also go to Ion Androutsopoulos, for his constructive discussions and meticulous comments regarding my work. I would also like to thank the rest of the members of my thesis examination committee, namely Dr. Constantinos Spyropoulos, Prof. Theodoros Kalaboukis, Prof. Emmanouel Giannakoudakis, Associate Prof. Emmanouel Giakoumakis, Associate Prof. Martha Sideri, and Dr. Iraklis Varlamis, for their constructive comments. Especially the help of Dr. Varlamis in the two final years of this work were determinant for the final outcome.

I have no words to thank my colleagues in the DB-NET team, former and present, namely Dimitris Mavroeidis, Christos Doulkeridis, Akrivi Vlachou, Nikos Salamanos, and Maria Halkidi. The continuous support that DB-NET members show to each other is something I will bear with me for the rest of my life.

Many more people over all these years have also played important roles in my research, influencing fruitfully the final outcome. I must, thus, thank Magdalini Eirinaki, Stratis Valavanis, Christos Pateritsas, and Yannis Batistakis for sharing with me their experience in the early stages of this work. Special thanks also to Kjetil Noervag. His professional way of thinking, his large experience in the field of computer science and the fruitful discussions I have had with him have armed me with confidence for my work.

None of this could have been achieved without the loving support of my family, and, of course, my wife Chryssa. I owe them great gratitude.

Last but certainly not least, I dedicate this work to the loving memory of my

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

father, Basileios Tsatsaronis, and my grand mother, Maria. Their presence taught me feelings that cannot be described with words, and their absence taught me the weaknesses of our human nature.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Contents

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

List of Tables

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

List of Figures

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Chapter 1

Introduction

The amount of text data has increased rapidly over the last two decades, especially with the advent and the wide use of the World Wide Web. Searching and organizing text information efficiently has become a very difficult task, mainly because of two reasons: (a) the diversity of the domains that texts refer to has increased dramatically, and (b) traditional methods for representing and processing text segments or documents employ mostly exact keyword matching.

The lexical ambiguity of most languages, like the English language, which is the language that this thesis focuses on, intensifies the problems arising from the aforementioned facts, with regards to computationally intensive tasks, like text classification and retrieval. Words and phrases can be found in different syntactic roles, while their meanings can also be different in various contexts, depending on the domain. The amalgamation of the several forms that lexical ambiguity can take, and which we will explain shortly in detail, has been shown to affect the performance of tasks like classification and retrieval [10, 51].

A natural solution to the problems arising from lexical ambiguity is to resolve it. However, automatically resolving lexical ambiguity by a computer program is not easy. This is the main reason that the problem of lexical ambiguity resolution is still a central problem in the areas of natural language processing and computational semantics in particular.

This thesis aims to explore semantic ambiguity, and proposes new Word Sense

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Disambiguation (WSD) algorithms to resolve it for the English language. In addition, this thesis focuses on the ways that semantic information from word thesauri, in our case WordNet [31], can be embedded into measures of text relatedness, and proposes a novel measure of text semantic relatedness, namely Omiotis. In order to explore the merits of embedding Omiotis into text related tasks, we present a series of experiments in several difficult tasks, like classification, retrieval, paraphrasing and document similarity.

The rest of this thesis is organized as follows:

- The remaining sections of chapter 1 present in detail the problem that we are trying to solve, set the boundaries of this work, and summarize the contributions.
- Chapter 2 analyzes the related work, with regards to previous WSD methods, representation of text with semantic networks, other semantic relatedness measures, as well as text applications of word thesauri like WordNet, including a discussion about the effect of embedding WordNet information into text retrieval.
- Chapter 3 presents our approaches for performing WSD and analyzes the experimental results in three benchmark collections for the task.
- Chapter 4 presents our novel measure of semantic relatedness, Omiotis.
- Chapter 5 analyzes the means of embedding Omiotis into well known text applications and also presents the experimental evaluation of Omiotis and discusses the experimental results.
- Chapter 6 concludes, by summarizing the contributions of this thesis and offering pointers to interesting future work.

1.1 Problem and Motivation

The classic models for representing documents are based on a vectorial representation of their terms. For decades now, one of the most important models of this classical

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

representation is the vector space model [98], which also constituted the basis for one of the most influential retrieval systems, namely SMART [97]. According to the vector space model, given a document d with i distinct terms $(t_1, ..., t_i)$, the vectorial representation of this document is a vector with i dimensions $\vec{d} = (w_1, ..., w_j)$, where $w_1, ..., w_j$ are either positive and non-binary, or binary, if the boolean vector model is used. In case the case that $(w_1, ..., w_j)$ represent weights of terms $(t_1, ..., t_i)$ in document d, a commonly used weighting scheme for the terms is the well known TF-IDF factor [110].

The vector space model allows the similarity between two documents (or between a document and a query in the IR paradigm) to be computed as the cosine of the angle between the two vectors. Though there have been many variations of the vector space model [5] since it was initially conceived, usually an assumption is made, known as *terms orthogonality*. According to this assumption, the vector space has as its orthocanonical base the term vectors $\vec{t_1} = (1, 0, 0..., 0), ..., \vec{t_i} = (0, 0, 0..., 1)$. This assumption implies that for a pair of documents, only their common terms affect their similarity, since any pair of term vectors $\vec{t_k}, \vec{t_m}$, where $t_k \neq t_m$, would produce a cosine value of zero (the two vectors are orthogonal). In the simplest case, the assumption leads to computing the similarity between documents based on *exact keyword matching*.

In what follows, we shall present some motivating examples of how exact keyword matching may affect the quality of certain text applications. In addition, we shall motivate the need for resolving lexical ambiguity in crucial tasks, such as IR. Primarily, let us consider the paraphrase recognition task, in which the target is to recognize whether a pair of sentences is approximately semantically equivalent. The following two sentences are an example of a paraphrase taken from the Microsoft Paraphrase Corpus [29]:

> The shares of the company dropped 14 cents.
>
> The organization's stock slumped 14 cents.

In this example, the noun *shares* of the first sentence is semantically vary similar to the noun *stock* of the second sentence. In fact, these two words are synonyms

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

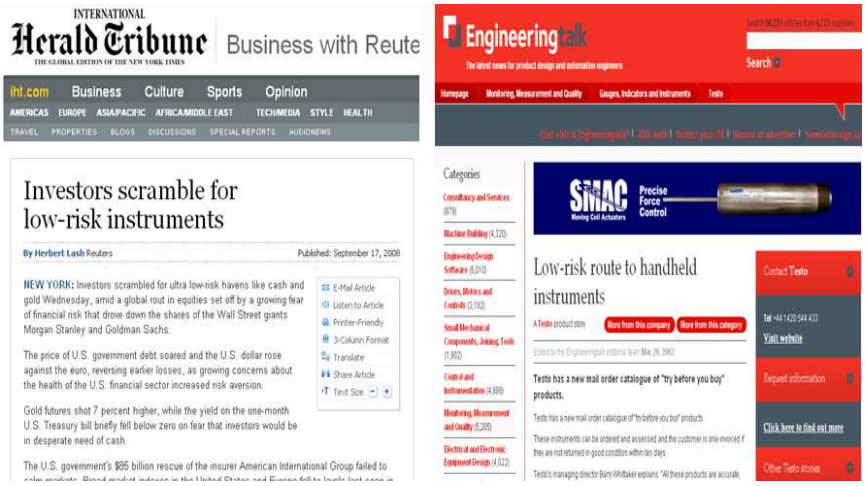

Figure 1.1: Top two Google results for query *Low-risk instruments*.

in WordNet. Furthermore, the verb *drop* of the first sentence, is a synonym of the verb *slump* of the second sentence. In addition, the nouns *company* and *organization* also bare a semantic similarity. The models based on exact keyword matching fail to recognize all of the aforementioned connections, residing in the absolute match of the number 14 and the noun *cent* to compute the similarity between these two sentences.

In another example, taken from the IR paradigm, let us consider a user searching for *low-risk instruments* in Google. This query pertains to information regarding legal documents and legislation about low-risk investment products. Submitting this query to Google, returns as the top two results the Web pages shown in figure 1.1. The first Web page (left) is an article from the *Herald Tribune* regarding investors, and it is certainly relevant to the user's needs. The second Web page (right) is a Web page that refers to engineering products (instruments) of a company named Testo. Clearly, in this second page the word *instruments* is used with another meaning than the one intended by the user, and of course this page is irrelevant to the user's needs. In this latter case, we observe how semantic lexical ambiguity can cause the performance of retrieval to deteriorate, if left unresolved.

These two examples, taken from the paraphrase recognition task and IR respectively, are representative of the caveats of exact keyword matching. The problem from this perspective is now reduced to the following. Traditional models of computing

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

document similarity: (a) do not consider semantic relations between documents, and (b) do not resolve lexical ambiguity. The result is a drop in the quality of the results in important text related tasks, like IR; as we shall see in the following, other tasks, such as text classification, are also affected.

Before we proceed our solution to this problem, let us first explore in more depth the different types of lexical ambiguity, and the ways it affects text applications. Lexical ambiguity can be roughly separated into two different types, namely *syntactic* and *semantic* ambiguity [13]. The former type of ambiguity stems from the fact that a word or phrase can have different syntactic roles. Consider the two following example sentences:

Oxides and hydroxides of metals and ammonia are included in *bases*.

He *bases* his claim on some observation.

In this example, *bases* has a different syntactic role in each sentence. In the first sentence, *bases* appears as the plural of the noun *base*, which in this case has a meaning taken from chemistry. In the second sentence, *bases* appears as a verb, with the meaning of *establish* or *ground*. The two sentences are an example of syntactic ambiguity.

In this thesis, we do not address the problem of syntactic lexical ambiguity. We only focus on the other large category of lexical ambiguity, semantic lexical ambiguity. In the following example, notice how differently the word *bank* is used in each sentence:

He cashed a check at the *bank*.

He sat on the *bank* of the river and watched the currents.

Examples like the above can be found in almost every English document, and the same applies to other natural languages. As a further example consider the following sentence, taken from [131]:

The young *page* put the goat in the *pen*.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

In this example, *page* and *pen* are very difficult to disambiguate, because they do not have their most common senses. Here, *page* takes the meaning of a boy who is employed to run errands, while *pen* means an enclosure for confining livestock.

Clearly, lexical ambiguity is a significant problem for natural language processing and text applications. This thesis presents new methods to cope with semantic lexical ambiguity in texts, by introducing new and efficient WSD algorithms. Furthermore, the thesis proposes a novel measure that embeds semantic information from WordNet into several critical text applications like text classification and text retrieval, and shows that semantic information can help to improve the respective tasks' results.

1.2 Considered Aspects and Proposed Solution

In an effort to improve the the results of text classification, retrieval and paraphrase recognition, where traditionally the vector based models that are based on exact keyword matching are used, in this work we aim at designing and implementing a measure that captures the semantic relatedness between words. Towards this direction, there are several possible options to consider, that have their individual advantages and disadvantages. Three basic research directions that are representative of the work in this area, are: (a) use of semantic information from word thesauri, (b) use of latent semantic analysis approaches, and (c) use of a language model. In this work, we focus only on the first direction, and, more specifically, we propose new WSD algorithms with the use of WordNet.

Furthermore, we design a new measure of semantic relatedness that capitalizes on the extracted semantic information and on the use of WordNet as a knowledge-base. Note, however, that in the remainder of this thesis we distinguish between semantic similarity and semantic relatedness, as discussed in the related bibliography [18]. The major difference between those two concepts, when a hierarchical word thesaurus is used, is that in the case of semantic similarity only the hierarchical relations are considered (i.e., hypernyms/hyponyms), while in the case of semantic relatedness, every non-hierarchical semantic relation can also be used. The measure we propose is a measure of semantic relatedness.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Overall, this thesis focuses on how we can address the limitations of exact keyword matching by embedding information from WordNet. More specifically, the presented work proposes novel solutions for the following: (1) extract lexical semantic information from text, (2) design and implement a word/text semantic relatedness measure that combines a thesaurus and the extracted lexical information, and (3) embed this measure into important text applications and evaluate its performance.

Regarding text pre-processing, unless stated otherwise, in the remainder of this thesis, the Stanford Tagger [116] is used for part-of-speech (POS) tagging, the Porter Stemmer is used for word stemming, and the TF-IDF formula [19] is used for term weighting in the vector space model. Finally, we use WordNet version 2.0, as our thesaurus. More information about WordNet, and the specific version is provided in appendix A. Note that concepts in the WordNet lexical database are represented by synsets (synonym sets), and that the terms *concepts* and *synsets* will be used interchangeably.

1.3 Summary of Contribution

Measuring the relatedness between two text segments in an automated manner is a difficult task. Text conveys semantics that are hard for a computer program to capture. Without doubt, a measure of relatedness between text segments must take into account both the lexical semantic relatedness between words and their significance in the text that they are found (e.g., their TF-IDF scores). Such a measure that combines both aspects may help in many tasks, such as text classification and retrieval. In this thesis we present a new approach for measuring the semantic relatedness between words based on their implicit semantic links. The approach does not require any type of training, since it exploits a word thesaurus, WordNet, in order to devise implicit semantic links between words. Based on this approach, we introduce a new measure of semantic relatedness between texts, which capitalizes on the semantic relatedness between individual words, and extends it to measure the relatedness between sets of words. We gradually validate our method: we first evaluate the performance of the semantic relatedness measure between individual words in four data sets and then

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

proceed with evaluating the performance of our method in measuring text-to-text semantic relatedness in three tasks. Experimental evaluation shows that the proposed method outperforms every other lexicon-based method of word semantic relatedness in the selected tasks and the tested data sets, and competes well against corpus-based approaches that require training. Finally, we show that the proposed measure can be successfully applied to more complex linguistic tasks (e.g. paraphrasing) and that it is able to capture the human notion of relatedness better than traditional lexical matching techniques.

The procedure of using semantic information from WordNet in text applications requires the design and implementation of a series of steps, most of which still constitute open research issues. Towards this direction, this thesis contributes in the following:

- **Word Sense Disambiguation:** We have developed four new methods of WSD, which we explain in detail in chapter 3, and which achieve state of the art results on three benchmark WSD data sets, namely Senseval-2 [83], Senseval-3 [108] and SemCor [71]. Three of these methods do not require any type of training and are dictionary-based approaches, relying only on WordNet. The fourth method constitutes an ensemble approach of dictionary-based methods to disambiguate words and requires the use of a training data set. Apart from one of the approaches, which focuses only on nouns, the other proposed methods can handle the disambiguation of any given word in unrestricted text.

- **Semantic Relatedness:** A novel measure of text semantic relatedness, Omiotis, is introduced, and explained in detail in chapter 4. Omiotis does not require training and is based on WordNet. The core element of the measure is SemanticRelatedness (SR), a measure for computing semantic relatedness between concepts in WordNet. SR is expanded to measure word-to-word relatedness and, eventually, to compute text-to-text relatedness, forming up the Omiotis measure. The usefulness of these measures is demonstrated with a series of experiments covering several different applications, from word-to-word relatedness to difficult text tasks. The experimental evaluation is analyzed in detail in

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

chapter 5.

- **Omiotis and SR Applications:** For some text applications, like text classification or text retrieval, embedding semantic information is not straightforward. In this direction, the thesis contributes by providing a novel semantic smoothing kernel for text classification and a novel generalized vector space model (GVSM) for text retrieval. Furthermore, we present additional interesting applications of these measures, in tasks like paraphrasing recognition and scholastic aptitude tests (SAT). The applications of SR and Omiotis are explained in section 5.1.

- **Novel Implementation:** The thesis also provides an implementation of SR and Omiotis, capable of handling the application of the measures to large data sets, like Reuters for classification and TREC for retrieval. As explained in appendix B, the implementation relies on a large database (around 600 GB of data), which indexes all the pairwise synset SR values for all WordNet synsets (11 billion combinations).

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Chapter 2

Background and Related Work

In this chapter we present previous work that is related to this thesis from the areas of WSD, semantic representation of text based on information from lexical databases or thesauri, and applications of text kernels and GVSM to text classification and text retrieval that use lexical information from dictionaries and/or thesauri.

2.1 Word Sense Disambiguation

Word Sense Disambiguation (WSD) aims to assign to every word of a document the most appropriate meaning (sense) among those offered by a lexicon or a thesaurus. WSD is important in natural language processing and in several applications, such as machine translation, speech processing and summarization. A wide range of WSD algorithms and techniques has been developed, utilizing machine readable dictionaries, statistical and machine learning methods, even parallel corpora. In [42] several approaches are surveyed; they address WSD either in a supervised manner, utilizing existing manually-tagged corpora, or with unsupervised methods, which sidestep the tedious stage of constructing manually-tagged corpora. Similar categorizations of WSD methods have been presented in [1] and [79]. In this thesis we follow Navigli [79], who distinguishes between supervised and unsupervised approaches, and also between knowledge-based and corpus-based approaches, with the former utilizing a

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

knowledge-base, like a lexical database or a thesaurus, whereas the latter rely on document collections instead. Based on this categorization, the relevant approaches to the ones we propose are knowledge-based approaches, both supervised and unsupervised.

Most of the approaches are traditionally evaluated in specific tasks of the Senseval initiative, which measure their ability to disambiguate all the words of texts (*English All Words* task of Senseval) or a targeted set of words (*English Lexical Sample* task).[1] The methods we propose are able to disambiguate all words in a given text segment, and, thus, we focus more on the corresponding task. The following three subsections discuss such approaches, giving greater emphasis to the methods that use a hierarchical word thesaurus, rather than a dictionary.

2.1.1 Knowledge-based Unsupervised Word Sense Disambiguation

Several WSD approaches capitalize on the fact that thesauri like WordNet offer important vertical (hypernym/hyponym) and horizontal (synonym, antonym, coordinate terms) semantic relations. Though early approaches of WSD, like the much influential method of Lesk [58], do not consider word thesauri, and rather use a dictionary to discover the correct senses through measuring the overlap of sense definitions and context words, the expansion of existing, and the development of new word thesauri has offered powerful knowledge that can be exploited. There have been approaches that try to expand the measurement of overlap of sense definitions to words of the thesaurus that are directly connected with the context words through strong semantic relations, like synonyms and hypernyms [6]. The performance of these approaches though, cannot compete with the results obtained from methods that use even more knowledge from rich thesauri like WordNet. A feature that distinguishes two of our proposed approaches for WSD, namely the Spreading Activation Networks approach that we introduced in [121] and a PageRank-based approach, both explained in detail in chapter 3, is the fact that we utilize all of the available semantic information

[1] `http://www.senseval.org/`

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

offered by WordNet. The state-of-the-art performance of these approaches on Senseval's data sets shows that the rich semantic information in WordNet can boost WSD performance.

The idea of using semantic relations from word thesauri in WSD is not new. Sussna [114] proposes a disambiguation algorithm, which assigns a sense to each noun in a window of context by minimizing a semantic distance function among their possible senses. The measure proposed is based on the assignment of weights to the edges in the WordNet noun hierarchy. To compute the weights, the is-a, has-part, is-a-part-of and antonyms relations between the noun senses are considered. Furthermore, the higher the level of the WordNet hierarchy, the greater is the conceptual distance that a semantic link between two senses suggests. Thus, Sussnas algorithm rewards semantic links between senses in lower levels of the WordNet noun hierarchy. This method has combinatory complexity due to the pair-wise computation of the semantic distance function for a given window of context.

Aggire and Rigau [3] introduce and apply a similarity measure based on conceptual density between noun senses. Their proposed measure is based on WordNet's is-a hierarchy and it measures the similarity between a target noun sense and the nouns in the surrounding context. For this purpose, they divide the WordNet noun is-a hierarchy into subhierarchies, so that each possible sense of the ambiguous noun belongs to exactly one subhierarchy. For each possible sense of the word to be disambiguated the measure returns the ratio of the area of the corresponding subhierarchy that is occupied by the context words (nouns only) to the total area occupied by the subhierarchy. The sense with the highest conceptual density (ratio) is assigned to the target word.

Banerjee and Pedersen [6] suggest an adaptation of the original Lesk algorithm in order to take advantage of the network of relations provided in WordNet. Rather than simply considering the glosses of the surrounding words in the sentence, the concept network of WordNet is exploited to allow for glosses of word senses related to the words in the context to be compared as well. Essentially, the glosses of surrounding words in the text are expanded to include glosses of those words to which they are related through relations in WordNet. They also suggest a scoring scheme such that

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

a match of n consecutive words in the glosses is weighted more heavily than a set of n one word matches.

More recent knowledge-based WSD approaches utilize even more semantic information from WordNet and can disambiguate words from all POS (nouns, verbs, adjectives and adverbs). An example of such an approach is the PageRank-based WSD method of Mihalcea et al. [70]. They use representation of WordNet as a graph, defining the vertices as synsets and the edges as the semantic relations connecting synsets. Adding some custom type edges in the same graph, they use this representation to construct synset graphs from text, and then execute the known PageRank algorithm to rank the synset vertices. However, they do not use weights on the edges, and they do not make use of all the semantic relations offered by WordNet 2.0. In chapter 3 we will show a modification of this algorithm, our proposed PageRank-based WSD method, relying on our novel semantic representation, and we prove experimentally that these modifications boost WSD performance on the Senseval data sets.

Navigli [81, 77, 78] presented the online implementation of Structural Semantic Interconnections (SSI-HITS), which constructs semantic graphs that connect all candidate senses and consequently ranks senses using the HITS algorithm. SSI-HITS is based on a measure that maximizes the degree of mutual interconnection among a set of senses, a variation of their former SSI WSD algorithm. The final sense selection for each term occurs after ranking the participating sense nodes in the constructed semantic graphs, using the HITS algorithm. Both SSI-HITS and the original SSI compare competitively as unsupervised WSD methods, though SSI performs better than its online implementation in Senseval 3, and can perform unrestricted text WSD. This method is similar to our PageRank-based based method (see chapter 3), but a different semantic network representation is used, as well as a different ranking algorithm for the nodes of the graph (HITS against PageRank).

There are also methods that propose a combination of unsupervised knowledge-based WSD algorithms to perform the task. Rigau et al. [93] propose a set of unsupervised knowledge-based heuristics and combine the disambiguation results with a weighted sum to produce the final decision for the disambiguation of a given word. Though Rigau et al. have applied it only to the disambiguation of genus terms of

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

two machine readable dictionaries (MRD), their method can also be applied to the disambiguation of unrestricted text, but Rigau et al. do not provide a means of tuning the weights of each heuristic. In some cases, knowledge-based methods and corpus-based methods are combined in an ensemble with simple voting mechanisms, to perform WSD, like for example in the work of Montoyo et al. [75]. Ensemble-based approaches rely on the hypothesis that WSD requires different types of knowledge sources to achieve high performance.

An ensemble methodology with higher performance than the method in [75] was proposed by Brody et al. [17], who use an ensemble of 4 unsupervised WSD methods (a Lesk-like extended gloss overlap method, a lexical chains method, the structural semantic interconnections method of Navigli [78], a distributional and WordNet similarity based method that learns predominant senses from raw text [65]) to boost overall performance and perform unrestricted text WSD. The method combines unsupervised methods, and uses their recommendations to reach a final decision. The overall method does not have a classifier on top of the unsupervised methods, and recommends senses and not WSD methods to be used. This ensemble approach of WSD methods is similar to the supervised ensemble WSD method that we propose (consult section 3.4), but it differs in that our ensemble recommends the WSD method that should be used to disambiguate each word occurrence, instead of proposing directly the sense for each word.

Another interesting unsupervised knowledge-based approach that utilizes measures of semantic relatedness or similarity to perform WSD is the approach by Sinha and Mihalcea [105]. They propose an unsupervised graph-based method for WSD, based on an algorithm that computes graph centrality of nodes in the constructed semantic graph. To measure the centrality of the nodes, they use the indegree, the closeness, the betweenness of the vertices in the graph, as well as PageRank. They also employ five known measures of semantic similarity or relatedness to compute the similarity of the nodes in the semantic graph, though the idea of using measures of semantic relatedness for performing WSD was initially employed by Patwardhan et al. [86]. The results of Sinha and Mihalcea are state-of-the art in the Senseval data sets, and their method is directly comparable to our PageRank method. The main

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

difference lies in the constructed semantic networks, as we will see in detail in section 3.3.2.

2.1.2 Knowledge-based Supervised Word Sense Disambiguation

Traditional supervised WSD methods use classifiers to predict the correct term sense from the context representation of the target word [87, 33, 54, 48]. In these methods a separate classifier learns to disambiguate the occurrences of each lemma (main form of a word). A feature vector is constructed for each lemma occurrence. The features are binary declaring the existence or absence of any other lemma in a given distance window from the target lemma occurrence. Supervised WSD becomes really hard to apply in this manner, since an ordinary document collection contains tens of thousands of different lemmas, leading to the construction of tens of thousands of classifiers. Furthermore, these methods require at least one occurrence of the target lemma in the training set to perform disambiguation, and are inapplicable to unrestricted text WSD. Moreover, their experimental evaluation is usually limited to a small set of target lemmas (at most 72 in the lexical sample Senseval tasks). Similar restrictions apply to WSD approaches that need at least one occurrence of each candidate sense in the manually annotated training corpora [137].

More formally, the basic idea of those methods is to use binary features $(F_1, F_2, ..., F_n)$, where each F_i suggests the existence or absence of a single word (or lemma) within a window of words to the left and the right of the target word, as for example shown in [87]. Though there is no global consensus on the selection of the appropriate features in the WSD task, the features must be chosen carefully. Their selection must be based upon three facts characterizing the domain of WSD: (a) Different parts of speech have different mean average polysemy, thus differing in the disambiguation difficulty. Using different features per POS, also depending on the number of senses a target word has may thus be desirable. (b) Many corpus-based methods, like the heuristic that always selects the first WordNet sense, are based on the frequencies of the senses of the target words in a corpus. This means these frequencies inside a

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

document and in the whole collection in general need to be considered. (c) The sense of a word occurrence can only be decided by examining its context, which was also the base thinking for early and successful Lesk-like WSD approaches.

In recently proposed supervised methods for WSD, the constraints that did not allow earlier supervised approaches to be applied to open text are relaxed, due to two main reasons: (a) more generic features are used, and (b) ensembles of approaches are used to increase coverage. Towards this direction, Mihalcea [67] presents the SenseLearner WSD system, which can perform unrestricted text WSD with very high accuracy. The system builds different semantic models for predefined different categories of terms, with a varying granularity of categories, and uses them to predict the correct sense per term. SenseLearner can disambiguate occurrences of only terms that have appeared at least once in the training corpus, or have been covered by the learned semantic models; Mihalcea uses the first WordNet sense heuristic for the other term occurrences.

In the approach of Kohomban and Lee [49] WSD is based on the construction of coarse grained semantic classes. Different classifiers learn to predict the correct semantic class for each term, and predictions are combined using a weighted majority voting scheme. The suggested semantic class is consequently mapped into finer grained senses based on heuristics. The accuracy of this method, which can perform WSD of unrestricted text, is comparable to fine grained WSD.

Hoste et al. [40] propose a WSD method that trains classifiers for each word-POS combination. To address the Senseval 2 *English all words task* they had to train 596 classifiers, one for each word-POS pair combination. The method can perform unrestricted text WSD, but has huge space complexity.

Le et al. [53] present a system that combines multiple classifiers to perform WSD. Their method is based on varying representations of each target term's context for each classifier. The classifiers' results are combined based on the Dempster-Shafer theory of evidence.

In contrast to the aforementioned approaches, our supervised ensemble approach described in section 3.4, decides on the appropriate base WSD method to be used for a term occurrence, and not on the correct sense, and since it uses unsupervised base

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

methods, it can be easily applied to unrestricted text. We provide a comparative analysis of the performance of the above methods in the Senseval 2 and 3 benchmark datasets, compared to our ensemble approach.

Finally, there are also some other approaches that can potentially disambiguate unrestricted text, but they are computationally infeasible for large amounts of text. For example, the method of Mihalcea and Moldovan [68] searches the Internet to find collocations of words in the glosses of the target terms and neighboring words to perform WSD. This is computationally expensive due to the large number of internet searches required.

2.1.3 Previous Use of SANs in WSD

Spreading Activation Networks (SANs) have already been used in information retrieval [26] and in text structure analysis [50]. Since the introduction of semantic networks by Quillian [90], several others [24, 15] have used semantic networks and spreading of activation in WSD, but those approaches required rather ad hoc hand-encoded sets of semantic features to compute semantic similarities. The most recent attempt to use SANs in WSD, overcoming the aforementioned drawback, is the work by Veronis and Ide [131].

Figure 2.1 illustrates how SANs were applied to WSD by Veronis and Ide. Let W_1 and W_2 be two words that co-occur (e.g., in a sentence or text) and which we want to disambiguate. They constitute the network nodes (word nodes) depicted in the *initial phase* of Figure 2.1; more generally, there would be n word nodes, corresponding to the n words of the text fragment. Next, all relevant senses of W_1 and W_2 are retrieved from a machine readable dictionary (MRD), and are added as nodes (sense nodes) to the network. Each word is connected to all of its senses via edges with positive weights (activatory edges). The senses of the same word (e.g. S_{11} and S_{12}) are connected to each other with edges of negative weight (inhibitory edges). This is depicted as *phase 1* in Figure 2.1. Next, the senses' glosses are retrieved, tokenized, and reduced to their lemmas (base forms of words). Stop-words are removed. Each gloss word (GW) is added as a node to the network, and is connected via activatory links to all sense

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

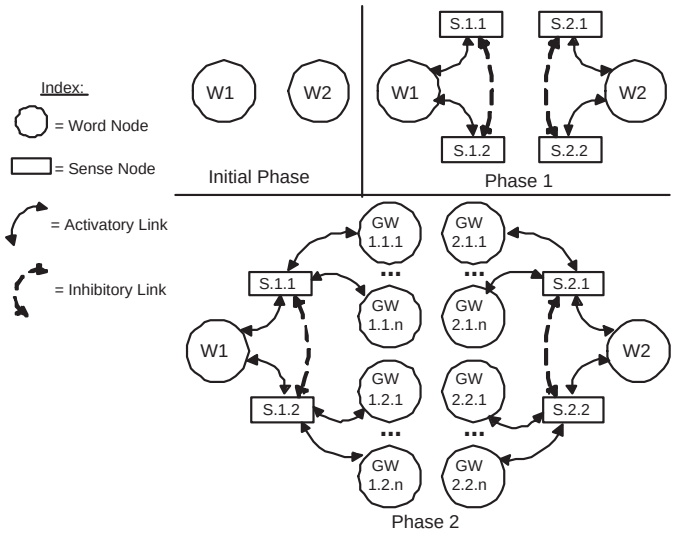

Figure 2.1: A previous method to generate SANs for WSD.

nodes that contain it in their glosses (*phase 2*). The possible senses of the gloss words are retrieved from the MRD and added to the network. The network continues to grow in the same manner, until nodes that correspond to a large part or the whole of the thesaurus have been added. Note that each edge is bi-directional, and each direction can have a different weight.

Once the network is constructed, the initial word nodes are activated, and their activation is spread through the network according to a spreading strategy, ensuring that eventually only one sense node per initial word node will have a positive activation value, which is taken to be the sense the algorithm assigns to the corresponding word. Note that this approach assumes that all occurrences of the same word in the text fragment we apply the algorithm to have the same sense, which is usually reasonable, at least for short fragments like sentences or paragraphs.

In section 3.3.1 we introduce a new spreading activation WSD algorithm, that is based on a novel semantic network construction methodology and a new spreading activation strategy. An important difference from the semantic network construction in [131] is that we also use sense-to-sense relations, while are not considered in the aforementioned approach. Furthermore, we do not make use of gloss-words. Experiments on WSD benchmarks show that our SAN method outperforms the method of

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Veronis and Ide.

2.2 Representation of Text using Semantic Networks

The use of a word thesaurus offers rich semantic relations between concepts and allows representing texts as semantic networks, initially introduced by Quilian [89]. The expansion of WordNet with semantic relations that cross parts of speech has added more possibilities of semantic network construction from text. Early approaches, (e.g., Veronis and Ide [131]) used the gloss words in the terms' definitions in order to build semantic networks from text. More recent approaches to semantic network construction from word thesauri, by Mihalcea et al. [70] and Navigli [78], utilize the semantic relations of WordNet instead. These methods outperformed previous methods that used semantic networks in the *all words* WSD tasks of Senseval 2 and 3 for English. In this thesis we adopt the semantic network construction method that we introduced in [121]. The method, which is explained in detail in section 3.3.1, utilizes all of the available semantic relations in WordNet 2.0 (Appendix A). Furthermore, we employ a novel weighting scheme for the edges connecting the sense nodes. WSD experiments show that the employed semantic network representation of text can be processed with several node ranking algorithms (e.g., activation spreading, PageRank), and that it outperforms previously proposed representations in WSD, like the one introduced by Mihalcea [70]. Furthermore, in chapter 4 we introduce our measure of semantic relatedness, which used the semantic network construction method in order to define semantic relatedness between concepts and words.

2.3 Measuring Semantic Relatedness Between Words

Semantic relatedness between words has been exploited in the past in text summarization, text retrieval and WSD [18]. The three most important factors of semantic graph based relatedness are: (a) length of the path connecting the senses, (b) the

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

depth of the senses in the used hierarchical thesaurus, and (c) the importance of the thesaurus edges involved. In section 4.1 we present a new measure of semantic relatedness (SR) [119] between WordNet concepts and we show how it is expanded to measure semantic relatedness between words and between texts (Omiotis). To the best of our knowledge, SR is the first measure of semantic relatedness that combines the three aforementioned factors. In general, the measures of semantic relatedness can be roughly classified in dictionary-based (also found in the bibliography as knowledge-based, or thesaurus-based), corpus-based and hybrid. Though SR and Omiotis belong clearly in the measures of the first category, in the experimental evaluation we compare them not only against all the state of the art dictionary-based measures, but also against some very important corpus-based and hybrid measures.

Regarding dictionary-based or hybrid measures that utilize a thesaurus, the measure of Agirre and Rigau in [2] computes the relatedness between sets of concepts based on the concepts' density and depth and on the length of the shortest path that connects them. However, Agirre and Rigau assume that all edges in the path are equally important. Resnik's [91] measure for pairs of concepts is based on the information content (IC) of the deepest concept that can subsume both. Measures proposed by Jiang and Conrath [45], Hirst and St-Onge [39], Leacock and Chodorrow [55], and Lin [59] were based on similar ideas. The reader may wish to consult Budanitsky and Hirst [18] for a detailed discussion of most of the aforementioned measures. All these measures use only the noun hierarchy, whereas our measure defines the semantic relatedness between any two terms, independently of their POS, utilizing all available semantic links offered by WordNet.

Some other, more recent, interesting measures of semantic relatedness, to which we also compare in our experiments, are: the measures of Jarmasz and Szpakowicz [44], who use Roget's thesaurus to compute semantic similarity; the LSA measure of Finkelstein et al. [32], who perform Latent Semantic Analysis (LSA) to capture text relatedness; the methods of Strube and Ponzetto [113] and Gabrilovich and Markovitch [34], who use Wikipedia to compute semantic relatedness; and finally the method of Mihalcea et al. [66], which is a hybrid method combining knowledge-based and corpus-based measures of text relatedness.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

2.4 Measuring Semantic Relatedness Between Text Segments

Most of the aforementioned measures of semantic similarity or distance or relatedness can be applied to pairs of words; their expansion to measure the relatedness between text segments requires an additional step. Though there are numerous string kernels than can measure the similarity between text segments by considering the string similarities of the words inside the texts [132], the approaches in the bibliography that can embed semantic similarity of words are few. A system that can achieve the thematic organization of Web documents (or text documents in general) by taking into account semantic information from a word thesaurus is THESUS [129], proposed by Varlamis et al. THESUS relied on Wu and Palmer's similarity measure [135] to cluster thematically a collection of documents. The noun hierarchy of WordNet was employed and experimental evaluation with two clustering algorithms (COBWEB and DBSCAN) showed that the clustering of THESUS produced higher F-Measure than the traditional VSM model with the cosine similarity measure. In our approach, we use THESUS to define Omiotis, our measure of semantic relatedness between text segments.

Besides THESUS, there are other approaches as well, that can be used to embed a semantic similarity measure of words in a semantic similarity measure of texts. Towards this direction, in an analogy to string kernels, there are semantic kernels that exploit knowledge bases like WordNet. Basili et al. [7, 10] define a semantic kernel that is applied to text classification. The measures of Wu and Palmer [135] and Lin [59] have been tested with Basili et al.'s kernel and the results presented in [10], that were obtained for a text classification task with Reuters, show that the defined kernel outperforms the linear kernel of Support Vector Machines (SVM). In section 5.1 we present a new kernel, based on the kernel in [10], by embedding SR as a measure of semantic relatedness. Experimental results show that in the same text classification task the kernel performs even better, due to the substitution of the relatedness measure.

Mihalcea et al. [66] propose another way to use measures of semantic similarity for

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

words to measure similarity between short text segments. Given two text segments T_1 and T_2, with n_1 and n_2 words respectively, they compute the following score between them:

$$sim(T_1, T_2) = \frac{1}{2}(\frac{\sum_{w \in T_1}(maxSim(w, T_2) * idf(w))}{\sum_{w \in T1} idf(w)} + \frac{\sum_{w \in T_2}(maxSim(w, T_1) * idf(w))}{\sum_{w \in T2} idf(w)}) \tag{2.1}$$

where $maxSim(w, T_1)$ is the maximum similarity found for the word w of T_2 with any word of the same part of speech from T_1, and $idf(w)$ is the inverse document frequency of word w in the collection. The THESUS formula that we use [129] is similar to the above, but considers similarity between all combinations of POS and normalizes to the number of terms in each text segment.

Other existing approaches for measuring text semantic similarity use Web resources. Cilibrasi and Vitanyi [21] propose the Google similarity distance, which is based on the page counts of word co-occurrences. Gabrilovich and Markovitch [34] and Strube and Ponzetto [113, 88] rely on Wikipedia to compute semantic relatedness between texts. Finally, there are also corpus-based approaches, like the the one proposed by Islam and Inkpen [43], combining corpus-based similarity and string similarity measures. These approaches, though offering high performance in many interesting tasks, have high computational cost due to the needed processing of huge Web sub-graphs or the required training and tuning. In contrast, Omiotis is a fast, unsupervised knowledge-based measure of semantic relatedness.

2.5 Word Thesauri and their Use in Text Applications

Word thesauri, like WordNet or Roget's International Thesaurus, constitute the knowledge-base for several text-related research tasks. WordNet has been used successfully as a knowledge base in the construction of Generalized Vector Space Models (GVSM) and semantic kernels for document similarity with application to text classification, like in the work of Basili et al.[7], or our previous work [63], and text retrieval,

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

like in the work of Voorhees [133], and the work of Stokoe et al. [112]. Furthermore, the idea of using a thesaurus as a knowledge base in text retrieval has also been proven successful in the case of cross language information retrieval, like in the CLIR system presented by Clough and Stevenson [22]. The design of a document similarity measure based on semantic kernels with application to information retrieval and/or text classification is a research challenge, since it involves investigating the impact of lexical ambiguity and WSD performance in those tasks. A short discussion follows that sums up the current trends in evaluating the impact of ambiguity in those tasks.

The most thorough investigation towards this direction, in information retrieval, is probably by Sanderson [101] who concludes that ambiguity in words can be found in many types, but new test collections are needed to realize the true importance of resolving ambiguity and using semantic relatedness measures and sense disambiguation in the text retrieval task. Earlier, Sanderson [100] reported that even 90% accurate WSD cannot guarantee retrieval improvement, though his experimental methodology was based only on randomly generated pseudowords of varying numbers of words. More precisely, he concluded that sense ambiguity is problematic for IR only when short queries are used. Similarly, Voorhees [133] reported a drop in retrieval performance when the retrieval model was based on WSD information. On the contrary, the construction of a sense-based retrieval model by Stokoe [112] improved performance, while several years before, Krovetz and Croft [51] had already pointed out that resolving word senses can improve searches requiring high levels of recall. More specifically, their results revealed that under certain circumstances, information about the senses of words may improve IR. Experiments on two document collections, CACM and TIMES showed that word senses provide a clear distinction between relevant and non-relevant documents, rejecting the null hypothesis that the senses of the words are not related to judgments of relevance. Also, they reached the conclusion that words being worth of disambiguation are either the words with uniform distribution of senses, or the words that have a different sense in the query from the most popular one.

From this small discussion it is evident that the impact of ambiguity in IR has raised a controversy that has been going on for almost two decades. From our point

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

of view, and since we have re-examined and verified part of the evaluation conducted by previous authors, like Krovetz and Croft [51] (consult Appendix D for our findings in five small IR collections), the opinion of Sanderson [101] is probably the most representative and descriptive of the situation: we do not have currently the data sets on which we can really evaluate the impact of lexical ambiguity in IR. Besides these conclusions, in section 5.1 we present a GVSM that embeds our SR measure of semantic relatedness into the text retrieval task, and we have conducted experimental evaluation on three TREC collections that shows slight improvement against the VSM model [117].

Regarding the use of semantic information from word thesauri in the text classification task, we shall briefly describe the previous relevant work on embedding WSD in document classification. The conclusions from the use of semantic information in text classification, in opposition to IR, are clear and compact, stating that semantic information (e.g. WSD information) improves document classification, especially when the training sizes are small. In [115], a WSD algorithm based on the general concept of extended gloss overlaps is used and classification is performed with an SVM classifier for the two largest categories of the Reuters-25178 collection and two IMDB movie genres.[2] It is demonstrated that, when the training set is small, the use of WordNet senses together with words improves the performance of the SVM classification algorithm; however for training sets above a certain size, the WSD approach is shown to have inferior performance compared to term-based classification. In that study, the semantic relations in WordNet were not exploited in the classification process. Although the WSD algorithm that was employed was not tested experimentally, its precision was estimated with a reference to [6], since the later work had a very similar theoretical basis. The experiments conducted in the latter study in Senseval 2 lexical sample data, show that the algorithm exhibits low precision (around 45%) and thus may introduce much noise that may deteriorate the overall performance in a classification task.

In [11], Bloehdorn and Hotho experiment with various mappings from words to senses including using the most frequent sense, as provided by WordNet, and using

[2] `http://www.imdb.com`

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

WSD based on context. Their approach is evaluated on Reuters-25178, OSHUMED and the FAODOC corpus, providing positive results. Their WSD algorithm has similar semantic information available as in the WSD algorithm proposed in [2], using only hypernyms.

In [95], Rosso et al. utilize the supervised WSD algorithm proposed in [74] in k-NN classification of the 20-newsgroups dataset. The WSD algorithm they employ is based on a Hidden Markov Model and is evaluated on Senseval 2 in the English *all words task*, achieving a maximum precision of around 60%. On the classification task of the 20-newsgroups dataset they report a very slight improvement in the error rate of the classification algorithm. WordNet's semantic relations are not exploited in the k-NN classification process.

Scott and Matwin [103] present an early attempt to incorporate semantics by means of hierarchical thesauri in the classification process, reporting negative results on the Reuters-21578 and DigiTrad collection. While no disambiguation algorithm is employed, hypernyms are used to extend the feature space representation.

In section 5.1 we present two new different ways of incorporating semantic information from WordNet to the text classification task, namely a GVSM model and a semantic smoothing kernel. Our experimental evaluation agrees with the reported conclusions of previous works: semantic information from a word thesaurus, like WordNet, can improve text classification performance, especially when the training sizes are small.

2.6 Generalized Vector Space Models

The Vector Space Model (VSM) assumes term orthogonality and, as explained in the introduction, this assumption misses much information regarding evidence of similarity among document vectors. Research has been conducted to formulate VSM generalizations (GVSM) that attempt to take into account the possible dependencies among terms in the VSM. In the remaining of this section we explain in detail the VSM and the GVSM.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

2.6.1 Vector Space Model

The VSM has been a standard model of representing documents in information retrieval for almost four decades [98, 128, 99, 5]. Let D be a document collection and Q the set of queries representing users' information needs. Let also t_i symbolize term i used to index the documents in the collection, with $i = 1, .., n$. The VSM assumes that for each term t_i there exists a vector $\vec{t_i}$ in the vector space that represents it. It then considers the set of all term vectors $\{\vec{t_i}\}$ to be the generating set of the vector space, thus the space basis. If each d_k (for $k = 1, .., p$) denotes a document of the collection, then there exists a linear combination of the term vectors $\{\vec{t_i}\}$ which represents each d_k in the vector space. Similarly, any query q can be modelled as a vector $\vec{q}$ that is a linear combination of the term vectors.

In the standard VSM, the term vectors are considered pairwise orthogonal, meaning that they are linearly independent. However, this assumption is unrealistic, since it requires lack of relatedness between any pair of terms, whereas the terms in a language often relate to each other. Provided that the orthogonality assumption holds, the similarity between a document vector $\vec{d_k}$ and a query vector $\vec{q}$ in the VSM can be expressed by the cosine measure given in equation 2.2.

$$cos(\vec{d_k}, \vec{q}) = \frac{\sum_{j=1}^{n} d_{kj} q_j}{\sqrt{\sum_{i=1}^{n} d_{ki}^2 \sum_{j=1}^{n} q_j^2}} \tag{2.2}$$

where d_{kj}, q_j are real numbers standing for the weights of term j in document d_k and query q respectively. A standard baseline retrieval strategy is to rank the documents according to their cosine similarity to the query.

2.6.2 Generalized Vector Space Model

Wong et al. [134] presented an analysis of the problems that the pairwise orthogonality assumption of the VSM creates. They were the first to address these problems by expanding the VSM. They introduced term to term correlations, which deprecated the pairwise orthogonality assumption, but they kept the assumption that the

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

term vectors are linearly independent, creating the first GVSM model.[3] More specifically, they considered a new space, where each term vector $\vec{t_i}$ was expressed as a linear combination of 2^n vectors $\vec{m_r}$, $r = 1..2^n$. The similarity measure between a document and a query then become as shown in equation 2.3, where $\vec{t_i}$ and $\vec{t_j}$ are now term vectors in a 2^n dimensional vector space, $\vec{d_k}$, $\vec{q}$ are the document and the query vectors, respectively, as before, $\acute{d_{ki}}, \acute{q_j}$ are the new weights, and $\acute{n}$ the new space dimensionality.

$$cos(\vec{d_k}, \vec{q}) = \frac{\sum_{j=1}^{\acute{n}} \sum_{i=1}^{\acute{n}} \acute{d_{ki}} \acute{q_j} \vec{t_i} \vec{t_j}}{\sqrt{\sum_{i=1}^{\acute{n}} \acute{d_{ki}}^2 \sum_{j=1}^{\acute{n}} \acute{q_j}^2}} \tag{2.3}$$

From equation 2.3 it follows that the term vectors $\vec{t_i}$ and $\vec{t_j}$ need not be known, as long as the dependencies between terms t_i and t_j are known. If pairwise orthogonality is assumed, the similarity measure is reduced to that of equation 2.2.

2.6.3 Semantic Information and GVSM

Based on the first GVSM model described earlier, it is evident that there are at least two basic directions for embedding term to term relatedness inside such a model: (a) compute semantic correlations between terms, or (b) compute frequency co-occurrence statistics of terms using large corpora. In this thesis we focus on the first direction to construct GVSM.

Several recent approaches have incorporated semantic information in VSM. In [63], we created a GVSM based on the use of noun senses, and their hypernyms from WordNet. We experimentally showed that this can improve text categorization. Stokoe et al. [112] reported an improvement in retrieval performance using a senses-based system. In [119] we show through another GVSM, presented in section 5.1, that semantic information and the use of a GVSM can improve the retrieval task. The latter approach differs from the aforementioned ones in that it expands the VSM model using the semantic information of a word thesaurus to interpret the orthogonality of terms and to measure semantic relatedness, instead of directly replacing terms with

[3] It is known from Linear Algebra that if every pair of vectors in a set of vectors is orthogonal, then this set of vectors is linearly independent, but not the inverse.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

senses, or adding senses to the model, as we shall explain in detail in chapter 4.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Chapter 3

Word Sense Disambiguation Using WordNet

In this chapter we analyze the contribution of this thesis in WSD. Four new WSD approaches are being proposed, that rely on the use of WordNet to disambiguate word occurrences in a given text. Experimental evaluation on the SemCor corpus, and the Senseval 2 and 3 *all English words* task show that the proposed methods produce state of the art results in the WSD bibliography. Regarding the selection of WordNet as the thesaurus of this thesis, there has been a lot of research work that compares experimentally the use of WordNet against other thesauri, like Roget's [76]. One of these studies is the one by Hale [64], who has implemented several semantic similarity measures using WordNet and Roget's thesaurus and has evaluated the measures on the Miller and Charles data set of word pairs similarities [72]. The results show that there are no major differences when using either of the two thesauri. Hence, we have chosen to use WordNet, since it is updated more frequently with new senses and relations than Roget's. Note however, that the aforementioned study also shows that semantic similarity measures used with WordNet can be also used with other thesauri. Thus, the measures and algorithms defined in the remaining of this thesis can be easily adapted to work with other thesauri as well.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

3.1 Evaluating WSD Performance

The typical evaluation methodology of WSD uses some measures from information retrieval, namely *precision*, *recall*, and the *F1* measure. In addition, *accuracy* and *coverage* are also used [79]. In the following, we will explain how these measures are defined in the context of WSD.

Given a data set $T = (w_1, ..., w_n)$ consisting of n word occurrences to be disambiguated, and given a function W that maps any $w_i \in T$ to a sense of this word from a dictionary D, then *coverage* C is defined as the percentage of the attempted answers (mappings) given by W, to the total number of the answers that should be provided (n):

$$C = \frac{\#answers}{n} \tag{3.1}$$

In the same context, *precision* P is defined as the percentage of correct answers provided by W, with regards to the total answers given by W:

$$P = \frac{\#correct}{\#answers} \tag{3.2}$$

Recall R is defined as the percentage of correct answers provided by W with regards to the total number of answers that should be returned:

$$R = \frac{\#correct}{n} \tag{3.3}$$

Precision and *Recall* are usually combined to a single measure, the $F1$ measure:

$$F1 = \frac{2PR}{P + R} \tag{3.4}$$

Finally, when the returned number of answers by W is equal to n (full coverage), it is common in the WSD bibliography [42, 79] to use the notion of *accuracy* of W, as the percentage of the correct answers with respect to n [70]; in this case precision and recall are identical to accuracy.

With regards to the data sets used to evaluate WSD systems, the Senseval initiative has conducted four series of evaluation exercises (Senseval 1, 2, 3 and SemEval),

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Senseval 2	**Nouns**	**Verbs**	**Adjectives**	**Adverbs**	**Total**
Mono.	260	33	80	91	464
Poly.	813	502	352	172	1839
Av. Poly.	4.21	9.9	3.94	3.23	5.37
Av. Poly. (Poly. only)	5.24	10.48	4.61	4.41	6.48
Senseval 3					
Mono.	193	39	72	13	317
Poly.	699	686	276	1	1662
Av. Poly.	5.07	11.49	4.13	1.07	7.23
Av. Poly. (Poly. only)	6.19	12.08	4.95	2.0	8.41
SemCor					
Mono.	16990	2584	9854	7831	37259
Poly.	70432	45117	24981	11878	152408
Av. Poly.	4.49	10.74	4.26	2.77	5.84
Av. Poly. (Poly. only)	5.33	11.29	5.55	3.93	7.02

Table 3.1: Occurrences of polysemous and monosemous words of WordNet 2 in Senseval 2, 3 and SemCor.

where the given data sets were manually annotated with the correct senses by human annotators using WordNet.[1] In each of these competitions there were many different tasks for several languages, with two of the most important ones being the *English all words* and the *English lexical sample* tasks. The data sets of the first task are widely used as benchmarks for WSD systems that can be applied to unrestricted text, while the data sets of latter task can be used with systems that only disambiguate of a set of few designated (target) words. Since we will be using the *English all words* exercises from Senseval 2 and 3 to evaluate the proposed algorithms, in table 3.1 we present the statistics of those data sets, including average polysemy of words, both with (Av. Poly.) and without (Av. Poly. (Poly. only)) taking into account monosemous words. We have also included an additional corpus that we adopt for evaluation, the SemCor corpus [71], which is larger than the data sets of the Senseval exercises and it is often used for the training of supervised WSD methods.

The table shows the number of monosemous (one sense given from WordNet)

[1] `http://www.senseval.org/`

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

and polysemous (more than one sense given from WordNet) word occurrences in the three data sets. Senseval 2 is easier to disambiguate than Senseval 3, as the average polysemy is larger in the latter. Adverbs are very easy to disambiguate and are usually excluded from the evaluation (i.e. Senseval 3 has only 13 adverb occurrences with average polysemy close to 1). The verb POS is the most difficult to disambiguate, since a typical verb has more than 8 different senses from WordNet.

Regarding the lower and upper bounds of WSD methods in those data sets, a straightforward lower bound is to select randomly a sense for each word occurrence. This disambiguation method would produce an accuracy of around 20% for Senseval 2 and SemCor, and 14% for Senseval 3. A reasonable upper bound, as stated in [79], would be the interannotator agreement or intertagger agreement (ITA), that is, the percentage of words tagged with the same sense by two or more human annotators. The interannotator agreement on coarse-grained (lexicons with few and clearly distinct senses for each lemma are used), possibly binary (two senses per lemma), sense inventories is calculated around 90% [35, 80], whereas on fine-grained, WordNet-style sense inventories, where there are many senses per lemma and which are often hard to distinguish, the inter-annotator agreement is estimated between 67% and 80% [20, 108, 82]. Similar findings have been reported, regarding the human performance in distinguishing among fine-grained and among coarse-grained senses of English word occurrences [37]. In the following sections, for each introduced method we present its performance on the data sets of table 3.1. The reader is requested to keep in mind the upper bound.

3.2 Compactness-based Word Sense Disambiguation

In this section we present an unsupervised WSD method, that we initially introduced in [62] and thoroughly evaluated in [63]. The algorithm can only disambiguate nouns and it is based on the intuition that adjacent terms extracted from a text document are expected to be semantically close to each other. Given a set of adjacent terms,

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

our disambiguation algorithm considers all the candidate sets of senses and output the set of senses that exhibits the highest level of semantic relatedness. Therefore, the main component of the algorithm is the definition of a semantic compactness measure for sets of senses.

Assuming that a document is represented by a set of senses, the semantic compactness measure that we introduce for WSD implies a similarity notion either among the senses of a sense set or between two sense sets. Its computation is based on the notion of Steiner Tree. Given a set of graph vertices, the Steiner Tree is the smallest tree that connects the set of nodes in the graph. The formal definition of the Steiner Tree is given below.

Definition 1 *Given an undirected graph $G = (V, E)$, and a set $S \subseteq V$, then the Steiner Tree is the minimal tree of G that contains all vertices of S.*

The use of the Steiner Tree in the formulation of semantic compactness stems from the fact that in most of the previous related approaches [45, 59, 91] the distance or similarity measure depend on the size of the shortest path that connects two concepts through a common ancestor in the hierarchy, or on the largest depth of a common ancestor in the hierarchy. The deepest common ancestor of a set of senses S will be denoted as $lca(S)$. Note, however, that in WordNet it is not always the case that a $lca(S)$ exists for S. In this case, the method cannot disambiguate the respective set of words. The definition of semantic compactness of a set of senses S follows.

Definition 2 *Given a Hierarchical Thesaurus (HT) O and a set of senses $S = \{s_1, ..., s_n\}$, where $s_i \in O$, the compactness of S is defined as the cost (weight) of the Steiner Tree of $S \cup lca(S)$, such that there exists at least one path from each s_i to the lca(S).*

In the definition above we include one path for every sense to the least common ancestor $lca(S)$. The reason for imposing such a restriction is that the distance between two concepts in a HT is not defined as the shortest path that connects them in the HT, but rather as the shortest path that goes through a common ancestor of a set of concepts S in which these two concepts belong to. Thus, it can be argued

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

that two concepts are connected only through a common ancestor and not through any other path in the HT. Including in the graph $lca(S)$ (and a path between every concept and the $lca(S)$) guarantees that a path connecting all pairs of concepts (in the sense discussed earlier) exists.

Although in general the problem of computing the Steiner Tree is NP-complete, the computation of the Steiner Tree (with the restriction imposed) of a set of concepts with their *lca* in a HT is computationally feasible and it is reduced to the computation of the shortest path from the *lca* to every concept of the set. Another issue, potentially adding excessive computational load, is the large number of combinations of possible sets of senses, when a term set of large cardinality is considered for disambiguation. In order to address this issue, we reduce the execution time by search the search space non-exhaustively with the use of a simulated annealing algorithm [25]. The proposed WSD algorithm can then be formulated as follows, given the definition of semantic compactness from definition 2.

Algorithm 1 requires a parameter T which is essentially the upper bound of the number of combinations of senses to be examined. In the experimental evaluation we set $T = 10000$. Another (implicit) parameter is the size of the window of adjacent word occurrences (W). Though a straightforward solution would be to truncate the text into sentences, experimental studies have shown that the size of the window is important in the disambiguation process [4]. Thus, in the experiments that follow, a parameter w declares the size of the considered windows.

Furthermore, since WordNet's noun hierarchy is actually a forest of nine trees with no common root the computation of the compactness is not always feasible, since for a given set of senses S, $lca(S)$ may not exist. To address this problem, we compute the windows compactness separately for each one of the nine trees of WordNet's noun hierarchy, and we sum the compactness scores we obtain to compute an overall compactness score. The number of allowed separate trees that the senses in S may belong to, is parameterized with L, where a value of $L = 0$ means that we will allow the computation of compactness only in the case that the senses, from the examined combinations, belong in the same WordNet noun tree. Respectively, a value of $L = 1$ means that the senses may belong to at most two different WordNet

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Algorithm 1 WSD-Semantic-Compactness(O,W,T)

Require: A word thesaurus O, a sequence of adjacent word occurrences W and a given positive constant T

Ensure: A mapping M of W to senses S or failure

1: MinCompactness = MAXIMUMPOSITIVEINTEGER
2: M=NULL
3: $S[i]$= select randomly a sense i of $w_i \in W$
4: tempCompactness = compactness(S)
5: **if** tempCompactness < MinCompactness **then**
6: MinCompactness = tempCompactness
7: M = assignments of senses to words of step 3
8: **end if**
9: **while** $T > 0$ **do**
10: S = nextCombinationOfSenses(W)
11: $T = T - 1$
12: tempCompactness = compactness(S)
13: **if** tempCompactness < MinCompactness **then**
14: MinCompactness = tempCompactness
15: M = assignment of senses to words of step 10
16: **else**
17: $p = e^{\frac{(tempCompactness - MinCompactness)}{T}}$
18: M = assignment of senses to words of step 10 with probability p
19: **end if**
20: **end while**
21: if $M \neq NULL$ return M else return failure

trees, and so on and so forth.

Experimental Evaluation

In figure 3.1 we present the results of algorithm 1 on the SemCor (Brown Corpus 1 and 2) and the Senseval 2 and 3 data sets for various settings of the parameters W and L. The white bars indicate precision and the black bars coverage. Only the nouns were considered in all data sets. The results are sorted in decreasing order of precision. The precision and coverage values reported do not take into account the monosemous nouns, but only the ambiguous ones. We can estimate, based on the examined corpora statistics, that the inclusion of the monosemous nouns would increase precision by 3%

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

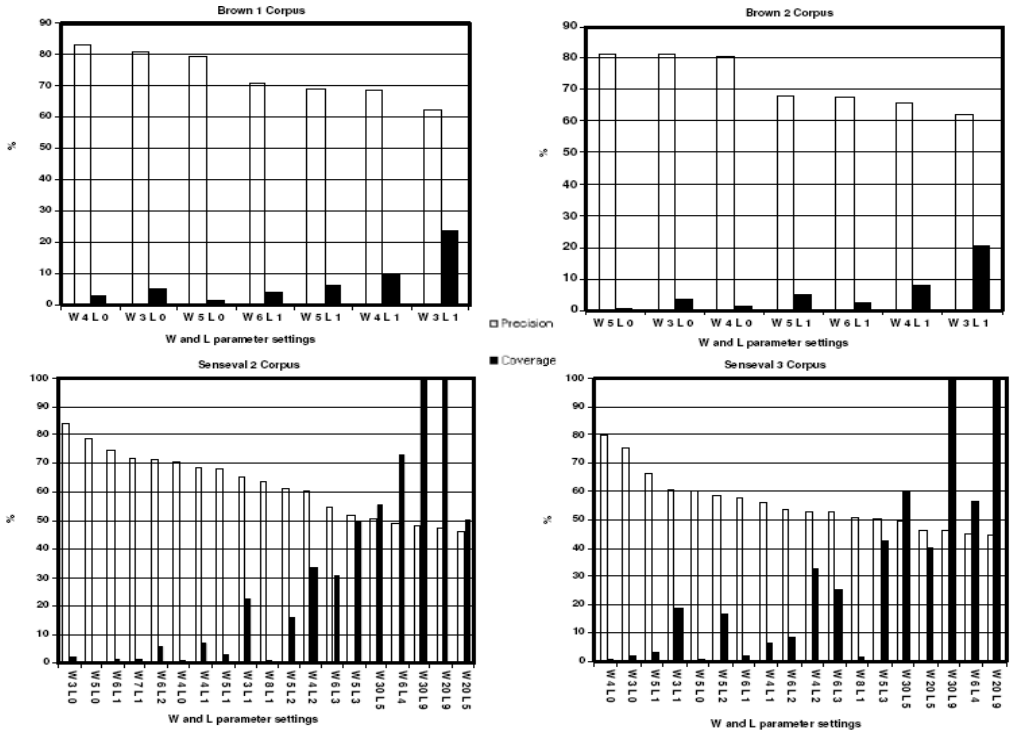

Figure 3.1: Performance of compactness in SemCor, Senseval 2 and Senseval 3 nouns.

to 4%; coverage would increase by almost 22%. Hence, compactness achieves precision greater than 80% with an associated coverage of more than 25%, if monosemous (i.e., non-ambiguous) nouns are also taken into account. Comparable experiments, including monosemous nouns, conducted in [3] reported a top precision result of 64,5% with an associated coverage of 86,2%. Similar experiments conducted in [114, 6, 74] resulted as well in lower precision than compactness. Comparing our approach to the state of the art WSD algorithms that were submitted to the *English All Words* task in Senseval 2[83] and 3[108], we observe that our approach can be configured to exhibit the highest precision for the noun POS. Of course the approach has two major problems: (a) it can only disambiguate nouns, and (b) precision drops dramatically if full coverage is achieved for large values of the l parameter. For this reason, we have extended the ideas of using semantic information from WordNet to disambiguate text by constructing semantic networks, and we have developed better algorithms for WSD, explained in the following sections, that solve the aforementioned problems.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

3.3 Word Sense Disambiguation Using Semantic Networks

In modern thesauri like WordNet 2.0, there are many semantic links between senses, which allow semantic networks to be constructed from text, as explained in section 2.2. Previous WSD approaches, however, have not considered the full range of semantic links between senses in a thesaurus. In [6] a larger subset of semantic relations compared to previous approaches was used, but antonymy, domain/domain terms and all inter-POS relations (e.g., linking noun senses to verb senses) were not considered. Similarly, other recent approaches like [70] and [77] consider a wide range of WordNet relations, but not the full range. Also, when constructing semantic networks, the weights of the edges are important, as pointed out in previous studies [114, 131].

In the next two sections we present two new WSD methods, SANs that are based on spreading of activation, and a PageRank-based approach. Both WSD systems are based on a semantic network construction method that we introduced in [121]. The edges in the semantic network are weighted and the full range of the WordNet semantic relations is considered. Experimental evaluation on the two Senseval data sets and SemCor shows that the two methods' performance is state of the art with respect to knowledge-based unsupervised methods. Moreover, the PageRank-based method has the highest reported accuracy compared to any other single (no ensembling) unsupervised knowledge-based method in all the tested data sets.

3.3.1 Word Sense Disambiguation with Spreading Activation Networks

The following WSD method we have developed requires no training and can disambiguate all words in a given text. WordNet is used to construct Spreading Activation Networks (SANs), initially introduced in WSD by Quillian [90]. The innovative points of this new WSD algorithm are: (a) it explores all types of semantic links, as provided by the thesaurus, even links that cross parts of speech, unlike previous knowledge-based approaches, which made use of mainly the "is-a" and "has-part" relations; (b)

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

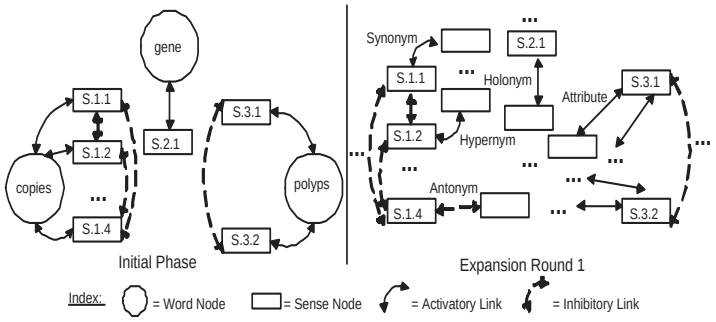

Figure 3.2: Our method to construct SANs.

it introduces a new method for constructing SANs for the WSD task; and (c) it introduces an innovative weighting scheme for the networks' edges, taking into account the importance of each edge type with respect to the whole network.

SAN Creation

For the construction of SANs we only consider the words of each sentence that are present in WordNet. We also assume that the words of the text have been tagged with their parts of speech (POS). For each sentence, a SAN is constructed as shown in figure 3.2. For simplicity, in this example we kept only the nouns of the input sentence, though the method disambiguates all parts of speech. The sentence is from the *d00* file of the Senseval 2 data set:

> "If both **copies** of a certain **gene** were knocked out, benign **polyps** would develop."

To construct the SAN, initially the word nodes, in our case the nouns *copies*, *gene* and *polyps*, along with their senses are added to the network, as shown in the *initial phase* of figure 3.2. The activatory and inhibitory links are then added, but after this point the SAN grows in a very different manner compared to Veronis and Ide. First, all the senses of the thesaurus that are directly linked to the existing senses of the SAN via any semantic relation are added to the SAN, along with the corresponding links, as shown in *expansion round 1* of figure 3.2. Every edge is bi-directional, since the semantic relations, at least in WordNet, are bi-directional (e.g.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

if S_1 is a hypernym of S_2, S_2 is a hyponym of S_1). In the next expansion round, the same process continues for the newly added sense nodes of the previous round. The network ceases growing when there is a path between every pair of the initial word nodes. Then the network is considered as *connected*. If there are no more senses to be expanded and the respective SAN is not connected, we cannot disambiguate the words of that sentence, losing in coverage. Note that when adding sense nodes, we use breadth-first search with a closed set, which guarantees we do not get trapped into cycles.

The Spreading Activation Strategy

The procedure above leads to networks with tens of thousands of nodes, and almost twice as many edges. Since each word is eventually assigned its most active sense, great care must be taken in such large networks, so that the activation is efficiently constrained, instead of spreading all over the network [26].

Our spreading activation strategy consists of iterations. The nodes initially have an activation level 0, except for the input word nodes, whose activation is 1. In each iteration, every node propagates its activation to its neighbors, as a function of its current activation value and the weights of the edges that connect it with its neighbors. We adopt the activation strategy introduced by Berger et al. [8], modifying it by inserting a new scheme to weigh the edges, which is discussed in the next section. More specifically, at each iteration p every network node j has an activation level $A_j(p)$ and an output $O_j(p)$, which is a function of its activation level, as shown in equation 3.5.

$$O_j(p) = f(A_j(p)) \tag{3.5}$$

The output of each node affects the next-iteration activation level of any node k towards which node j has a directed edge. Thus, the activation level of each network node k at iteration p is a function of the output, at iteration $p-1$, of every neighboring node j having a directed edge e_{jk}, as well as a function of the edge weight W_{jk}, as shown in equation 3.6. Although this process is similar to the activation spreading of feed-forward neural networks, the reader should keep in mind that the edges of SANs

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

are bi-directional (for each edge, there exists a reciprocal edge). A further difference is that no training is involved in the case of SANs.

$$A_k(p) = \sum_j O_j(p-1) \cdot W_{jk} \tag{3.6}$$

Unless a function for the output O is chosen carefully, after a number of iterations the activation floods the network nodes. We use the function of equation 3.7, which incorporates fan-out and distance factors to constrain the activation spreading; τ is a threshold value.

$$O_j(p) = \begin{cases} 0 & , \text{ if } A_j(p) < \tau \\ \frac{F_j}{p+1} \cdot A_j(p) & , \text{ otherwise} \end{cases} \tag{3.7}$$

Equation 3.7 prohibits the nodes with low activation levels from influencing their neighboring nodes. The factor $\frac{1}{p+1}$ diminishes the influence of a node to its neighbors as the iterations progress (intuitively, as "pulses" travel further). Function F_j is a fan-out factor, defined in equation 3.8. It reduces the influence of nodes that connect to many neighbors.

$$F_j = (1 - \frac{C_j}{C_T}) \tag{3.8}$$

C_T is the total number of nodes, and C_j is the number of nodes directly connected to j via directed edges from j.

Assigning Weights to Edges

In information retrieval, a common way to measure a token's importance in a document is to multiply its term frequency in the document (TF) with the inverse (or log-inverse) of its document frequency (IDF), i.e. with the number of documents the token occurs in. To apply the same principle to the weighting of SAN edges, we consider each node of a SAN as corresponding to a document, and each type of edge (each kind of semantic relation) as corresponding to a token.

Initially each edge of the SAN is assigned a weight of -1 if it is inhibitory (edges representing antonymy and competing senses of the same word), or 1 if it is activatory (all other edges). Once the network is constructed, we multiply the initial weight w_{kj}

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

of every edge e_{kj} with the following quantity:

$$ETF(e_{kj}) \cdot INF(e_{kj}) \tag{3.9}$$

ETF, defined in equation 3.10, is the edge type frequency, the equivalent of *TF*. It represents the percentage of the outgoing edges of k that are of the same type as e_{kj}. When computing the edge weights, edges corresponding to hypernym and hyponym links are considered of the same type, since they are symmetric. The intuition behind *ETF* is to promote edges whose type is frequent among the outgoing edges of node k, because nodes with many edges of the same type are more likely to be hubs for the semantic relation that corresponds to that type.

$$ETF(e_{kj}) = \frac{|\{e_{ki}|type(e_{ki}) = type(e_{kj})\}|}{|\{e_{ki}\}|} \tag{3.10}$$

The second factor in equation 3.9, defined in equation 3.11, is the inverse node frequency (*INF*), inspired by *IDF*. It is the frequency of e_{kj}'s type in the entire SAN.

$$INF(e_{kj}) = \log \frac{N+1}{N_{type(e_{kj})}} \tag{3.11}$$

N is the total number of nodes in the SAN, and $N_{type(e_{kj})}$ is the number of nodes that have outgoing edges of the same type as e_{kj}. As in *IDF*, the intuition behind *INF* is that we want to promote edges of types that are rare in the SAN.

The WSD Algorithm

Our WSD algorithm consists of the steps shown in algorithm 2, given a POS-tagged text, a designated set of parts of speech to be disambiguated, and a word thesaurus. Note that during the spreading of the activation, activation spreads iteratively until all nodes are inactive.[2] For every word node, eventually the sense node with the highest activation is kept. If there is more than one sense node with this property per word, we select randomly. This never occurred in our experiments.

[2]In equation 3.7, $O_j(p)$ is bounded, because as p increases it approaches 0. Eventually, all nodes become inactive.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Algorithm 2 WSD-SAN(O,T,P)

Require: A word thesaurus O, a part of speech tagged text T, and a set of designated parts of speech for disambiguation P.
Ensure: A mapping M of all word occurrences W of T to senses S or failure

```
1: Sen = fragmentTextIntoSentences(T)
2: for all Sen[i] do
3:    W[i] = keepWordsOfDesignatedPOS(Sen[i],P)
4: end for
5: for all W[i] do
6:    G = constructSANForSetOfWords(W[i],O)
7:    if G is a connected graph then
8:       spreadActivation(G)
9:       S[i] = getMostActiveSenseNodesForWords(G,W[i])
10:   else
11:      S[i] = NULL
12:   end if
13: end for
14: M = map(W,S)
15: return M
```

Experimental Evaluation

We evaluated our algorithm on Senseval 2, 3 and SemCor. Prior to presenting the results from all data sets, and for reasons of straightforward comparison with other methods, we will first present a comparative evaluation in Senseval 2. We experimented with all parts of speech, to be compatible with all published results of Senseval 2 [83].

In order to compare our WSD method to the method of Veronis and Ide [131], we implemented the latter and evaluated it on Senseval 2. We also include in the comparison the baseline for unsupervised WSD methods, i.e., the assignment of a random sense to each word. For the baseline, the mean average of 10 executions (random assignments) is reported. Moreover, in order to evaluate the possibility of including glosses in our method, instead of only synset-to-synset relations, we implemented a hybrid method which utilizes both, by adding to our SANs the gloss words of the synsets along with their senses, similarly to the method of Veronis and Ide (section 2.1.3). For the purposes of this implementation, as well as for the

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

	Words		*SAN*	*SAN Glosses*	*SAN*	*Baseline*	*Best Unsup.*	*Pagerank*
	Mono	*Poly*	*Synsets*	*Veronis and Ide*	*Syn.+Glosses*		*Senseval 2*	*Mihalcea*
File 1 (d00)	103	552	0.4595	0.4076	0.4396	0.3651	UN/A	0.4394
File 2 (d01)	232	724	0.4686	0.4592	0.4801	0.4211	UN/A	0.5446
File 3 (d02)	129	563	0.5578	0.4682	0.5115	0.4303	UN/A	0.5428
Overall	464	1839	0.4928	0.4472	0.4780	0.4079	0.4510	0.5089

Table 3.2: Overall and per file accuracy on the Senseval 2 data set.

implementation of the original method of Veronis and Ide, we used the Extended WordNet [73], which provides the POS tags and lemmas of all WordNet 2 synset glosses. In the comparison, we also include the results presented by Mihalcea et al. [70]. Their method is an unsupervised knowledge-based WSD method, evaluated on Senseval 2; the method uses thesauri-generated semantic networks, along with Pagerank for their processing. We also report the accuracy of the best reported unsupervised method that participated in the Senseval 2 "*English all words*" task, presented in [60].

Table 3.2 presents the accuracy of the six WSD methods, on the three files of Senseval 2. The presented accuracy corresponds to full coverage, and hence recall and precision are both equal to accuracy. The results in Table 3.2 suggest that our method outperforms that of Veronis and Ide, the hybrid method, and the random baseline. Moreover, our method achieved higher accuracy than the best unsupervised method that participated in Senseval 2, and overall slightly lower accuracy than the reported results of Mihalcea et al. [70].

Figure 3.3 shows the corresponding overall results for the four methods we implemented, when accuracy is computed only on polysemous words, i.e. excluding trivial cases, along with the corresponding 0.95 confidence intervals. There is clearly a statistically significant advantage of our method (Synsets) over both the baseline and the method of Veronis and Ide. Adding WordNet's glosses to our method (Synsets+Glosses) does not lead to statistically significant difference (overlapping confidence intervals), and hence our method without glosses is better, since it is simpler and requires lower computational cost, as explained later. The decrease in

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

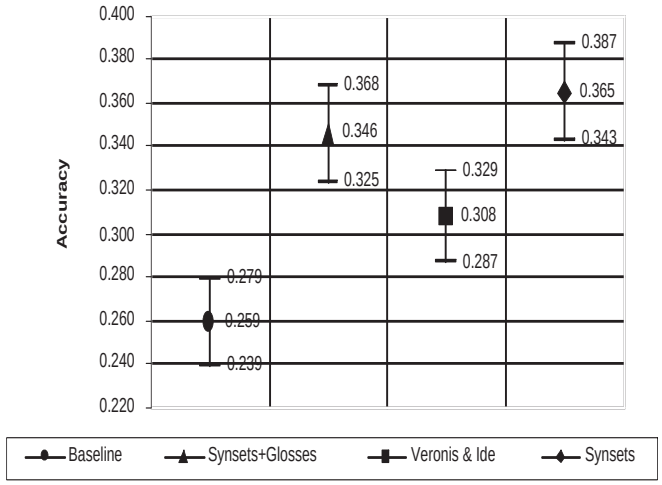

Figure 3.3: Accuracy on polysemous words and the respective 0.95 confidence intervals.

performance when adding glosses is justified by the fact that many of the glosses' words are not relevant to the senses the glosses express, and thus the use of glosses introduces irrelevant links to the SANs.

Figure 3.3 does not show the corresponding results of Mihalcea et al.'s method, due to the lack of corresponding published results; the same applies to the best unsupervised method of Senseval 2. We note that in the results presented by Mihalcea et al., there is no allusion to the variance in the accuracy of their method, which occurs by random assignment of senses to words that could not be disambiguated, nor to the number of these words. Thus no direct and clear statement can be made regarding their reported accuracy. In Figure 3.4 we compare the accuracy of our method against Mihalcea et al.'s on each Senseval 2 file. In this case we included all words, monosemous and polysemous, because we do not have results for Mihalcea et al.'s method on polysemous words only; the reader should keep in mind that these results are less informative than the ones of Figure 3.3, because they do not exclude monosemous words. There is an overlap between the two confidence intervals for 2 out of 3 files, and thus the difference is not always statistically significant.

Regarding the best unsupervised method that participated in Senseval 2, we do

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

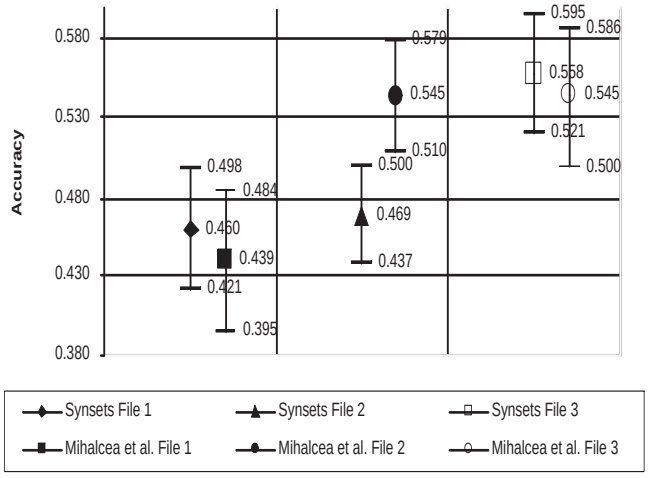

Figure 3.4: Accuracy on all words and the respective 0.95 confidence intervals.

POS	*Senseval 2*	*Senseval 3*	*SemCor*
NOUN	0.5396	0.5078	0.5086
VERB	0.3177	0.3641	0.3512
ADJECTIVE	0.5902	0.5804	0.5576
ALL	0.4928	0.4676	0.4859

Table 3.3: Overall and per POS accuracy of SANs in the three data sets.

not have any further information apart from its overall accuracy, and therefore we rest on our advantage in accuracy reported in table 3.2. Finally, we note that to evaluate the significance of our weighting, we also executed experiments without taking it into account in the WSD process. The accuracy in this case drops by almost 1%, and the difference in accuracy between the resulting version of our method and the method of Veronis and Ide is no longer statistically significant, which illustrates the importance of our weighting. We have also conducted experiments in Senseval 3, where similar results with statistically significant differences were obtained: our method achieved an overall accuracy of around 46%, while Ide and Veronis achieved $39,7\%$. In table 3.3 we present the results of the SANs method in Senseval 2 and 3, as well as in SemCor. The table shows the performance of the method per POS.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Measurement	SAN Synsets	SAN Glosses Veronis and Ide	SAN Synsets+Glosses
Nodes/Net.	10,643.74	6,575.13	9,406.04
Edges/Net.	13,164.84	34,665.53	37,181.64
Pulses/Net.	166.93	28.64	119.15
Sec./Net.	13.21	3.35	19.71

Table 3.4: Average actual computational cost.

Complexity and Actual Computational Cost

Regarding the complexity and the computational costs of SANs, let k be the maximum branching factor (maximum number of edges per node) in a word thesaurus, l the maximum path length, following any type of semantic link, between any two nodes, and n the number of words to be disambiguated. Since we use breadth-first search, the computational complexity of constructing each SAN (network) is $O(n \cdot k^{l+1})$. Furthermore, considering the analysis of constrained spreading activation in [94], the computational complexity of spreading the activation is $O(n^2 \cdot k^{2l+3})$. The same computational complexity figures apply to the method of Veronis and Ide, as well as to the hybrid one, although k and l differ across the three methods. These figures, however, are worst case estimates, and in practice we measured much lower computational cost. In order to make the comparison of these three methods more concrete with respect to their actual computational cost, table 3.4 shows the average numbers of nodes, edges, and iterations per network (sentence) for each method. Moreover, the average CPU time per network is shown (in seconds), which includes both network construction and activation spreading. The average time for the SAN Synsets method to disambiguate a word was 1.37 seconds. Table 3.4 shows that our method requires less CPU time than the hybrid method, with which there is no statistically significant difference in accuracy; hence, adding glosses to our method clearly has no advantage. The method of Veronis and Ide has lower computational cost, but this comes at the expense of a statistically significant deterioration in performance. Mihalcea et al. provide no comparable measurements, and thus we cannot compare against them; the same applies to the best unsupervised method of Senseval 2.

3.3.2 PageRank-based Word Sense Disambiguation

In order to investigate further the potential of the semantic representation that we introduced earlier, we designed another WSD algorithm that uses the representation of the SANs method and processes the constructed networks with PageRank. The PageRank formula that we used is a simple variation of the original PageRank equation, which takes into account edge weights as well. This variation was first introduced by Mihalcea et al. in [70]. Equation 3.12 shows the original PageRank formula and equation 3.13 shows its weighted variation that we use to process the networks. $S(V_i)$ (and $WS(V_i)$ respectively) is the PageRank value of vertex V_i, d is the damping factor, $Out(V_j)$ is the number of outgoing links from vertex V_i and w_{ij} is the weight of the edge connecting vertices V_i and V_j.

$$S(V_i) = (1-d) + d \sum_{j \in In(V_i)} \frac{S(V_i)}{|Out(V_j)|} \tag{3.12}$$

$$WS(V_i) = (1-d) + d \sum_{V_j \in In(V_i)} \frac{w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \tag{3.13}$$

Algorithm 2 can then be easily modified to process the constructed networks with equation 3.13, instead of spreading of activation. As a dumping factor (d) we set 0.85, as in the original formula by Brin and Page [16]. After the PageRank values stabilize, the sense nodes with the highest PagerRank scores for each target word are selected to disambiguate each word occurrence. The complexity of the network construction was discussed in the previous section.

Experimental Evaluation

To evaluate the performance of this new PageRank-based WSD algorithm, we experimented with Senseval 2, 3 and SemCor. Table 3.5 shows the accuracy of our method for all POS in the three data sets. The proposed PageRank-based algorithm surpasses (with statistical significance at the 0.95 confidence level) both our SANs method [121] and the method of Mihalcea et al. [65]. To the best of our knowledge, this method is currently the best performing unsupervised knowledge-based method,

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

POS	*Senseval 2*	*Senseval 3*	*SemCor*
NOUN	0.6439	0.5728	0.6137
VERB	0.3271	0.4179	0.4216
ADJECTIVE	0.5810	0.5459	0.6355
ALL	0.5475	0.5109	0.5633

Table 3.5: Overall and per POS accuracy of PageRank in the three data sets.

when no ensembles of several methods are used.

3.4 Ensemble of WSD Methods

The interannotator agreement in the Senseval competitions ranges from 67% to 80% [79]. As shown in the previous sections, the state of the art in unsupervised knowledge-based WSD methods reaches up to approximately 55%. In order to close the performance gap between automated WSD algorithms and theoretical human performance (interannotator agreement) several supervised WSD methods have been proposed in the past, that utilize machine learning techniques to learn the correct sense of each word occurrence, as discussed in section 2.1.2.

In this section we propose a new supervised WSD method based on an ensemble of unsupervised knowledge-based methods. More precisely, in order to address the *knowledge acquisition bottleneck* problem, without the need for extensive training, which typically requires manual annotation effort, we combine several Wordnet-based WSD approaches, which do not require training. Training is required only for deciding which method to trust per word occurrence. Consequently the WSD problem for a word occurrence splits into two separate decision problems: a) find the degree of trust for each WSD method per term occurrence based on a set of term features and, b) decide on the method that will disambiguate the target term occurrence.

For the trustfulness problem we use one Support Vector Machine (SVM) classifier for each WSD method. For a word occurrence, the classifier examines a small set of lexical and syntactic features and decides how much we should trust the respective WSD method (the positive class stands for *trust*, while the negative one for *do not*

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

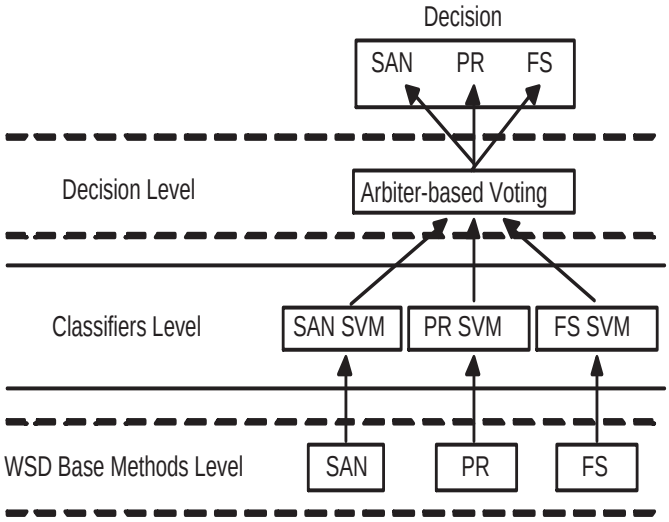

Figure 3.5: Overall System Organization.

trust) for the disambiguation of the particular word occurrence. We combine the classifiers' decisions using a simple arbiter-based voting mechanism, and we choose the prevailing method to disambiguate the given word occurrence.

A major advantage of this method is its flexibility, since the modules of each level (see figure 3.5) can be enhanced or replaced by others, as we show in the experiments: a) the base WSD methods can be either supervised or unsupervised, b) the number of the base WSD methods and the respective classifiers can be increased or decreased, c) different types of classifiers (e.g., SVM, Maximum Entropy) can be employed.

System Overview

The proposed WSD system combines the merits of three WordNet-based WSD methods in order to raise the overall WSD accuracy. The overall system organization is shown in figure 3.5, where the two dotted levels do not have any need for training. The system can disambiguate any word occurrence, provided the word occurs in the lexicon, without restrictions (i.e., in context or part of speech). Each WSD method in the lowest layer gives a candidate sense for the term. A set of classifiers, in the middle layer, examines specific features of the term and produces a binary decision

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

POS	*Senseval 2*	*Senseval 3*	*SemCor*
NOUN	0.711	0.6972	0.7851
VERB	0.4253	0.524	0.6131
ADJECTIVE	0.6759	0.6724	0.813
ALL	0.637	0.613	0.7425

Table 3.6: Overall and per POS accuracy of FS in the three data sets.

(*trust* or *do not trust*) for each WSD method. The final decision is drawn at the top layer, where the prevailing WSD method is selected from the ensemble using a simple arbiter-based voting formula that does not require any type of training.

The details of the classifiers' training, the selected set of features and the ensembling mechanisms are presented later. The three lexicon-based methods we employ are the Spreading Activation Networks (SAN) introduced in section 3.3.1, a simple baseline method that always selects the first (most frequent) sense from WordNet (FS) and is usually the baseline for supervised WSD methods, and finally the PageRank-based method (PR) introduced in section 3.3.2, which constitutes a combination of SAN and the method presented in [70].

Knowledge-based WSD Methods

The main requirements for the suggested WSD system, in order to achieve high disambiguation performance are: (1) the base methods should provide state of the art performance, and (2) they should have low level of pairwise inter-agreement in their disambiguation result, so that each method can act as a complement to the rest. In simple words, performance is high when at least one of the WSD methods in the lower layer gives the correct sense, and when the WSD methods often disagree. In this section we explain why the selected three WSD methods meet both conditions. Prior to that, we explain the FS method in detail. This simple, yet powerful heuristic, is used as a baseline in supervised WSD. As McCarthy et al. [65] noticed, FS is so powerful because the distribution of word senses is often skewed; the most probable sense is dominant. For example, in 63.7% of the word occurrences of Senseval 2 and 61.3% of Senseval 3, the correct sense is the first WordNet sense. Table 3.6 shows the


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

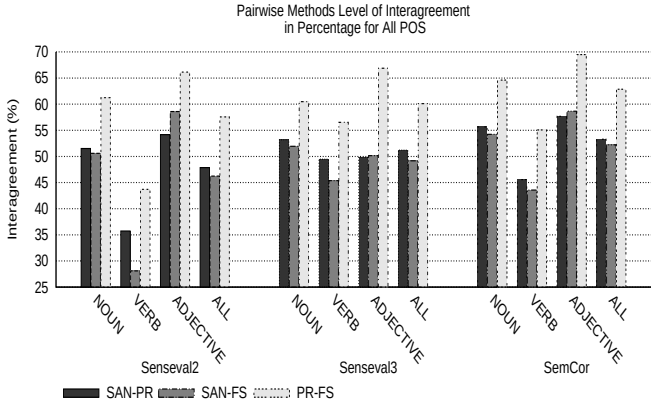

Figure 3.6: Pairwise methods inter-agreement in sense level.

performance per POS of the FS method in the three data sets. Clearly, FS surpasses both the SANs and the PR method in performance. It relies, however, on statistics from annotated corpora that the WordNet developers analyzed to rank the senses by decreasing frequencies, and in that sense FS is supervised, whereas SANs and PR are unsupervised.

All three methods, however, have very high performance on the tested data sets. An analysis of the interagreement of the three methods is shown in figure 3.6, per POS and data set. Interagreement has been computed for all pairwise combinations of the methods. A successful combination of methods should have low interagreement rates (below 80%) to improve the overall WSD performance, since an ensemble of WSD methods, as analyzed in [33], brings relatively little performance improvement over individual classifiers when the classifiers have a very high inter-agreement rate. Indeed, the selected methods have low levels of inter-agreement for all POS.

Classifiers Ensemble

First, we train one classifier for each base WSD method, using a set of sense-tagged training term occurrences and a limited set of grammatical and syntactic features for each term. We consequently merge the classifiers' decisions (confidence scores) using a voting mechanism. The details of the classification features, the classifiers, and the voting mechanism follow.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Feature	NPOS	VPOS	AJPOS	AVPOS	SN	CWA	CPA	CPT	CLT	WFTS
Descr.	0,1	0,1	0,1	0,1	# Target word's senses	% correct word sense ass.	% correct POS sense ass.	% correct POS triad ass.	% correct lemma triad ass.	Target word's FOC

Table 3.7: Selected set of features.

We will first discuss the selected feature set. Previous work [53] has shown that WSD performance improves when several different categories of features are combined. Consequently, we aimed at a small and simple feature set that combines different kinds of features, shown in table 3.7. The first four features ($NPOS$, $VPOS$, $AJPOS$ and $AVPOS$) are binary and represent the POS of the word occurrence (the one to be disambiguated). *Senses Number* (SN) is an integer indicating the polysemy of the target word. *Correct Word Assignment* (CWA) and *Correct POS Assignment* (CPA) are both real numbers in [0..1] representing the percentage of correct sense assignments to the occurrences of the target word and the POS of the target word in the training set for the base disambiguation method the classifier corresponds to. With regards to the *Collocated Lemma Triads* (CLT) and *Collocated POS Triads* (CPT), let (L_{T-1}, L_T, L_{T+1}) be the ordered triple of the target lemma, the previous and the next lemma; and let $N_{(L_{T-1},L_T,L_{T+1})}$ be the number of occurrences of this triple in the training set.[3] If $C_{(L_{T-1},L_T,L_{T+1})}$ is the number of correct word-sense assignments to the target lemma in the training set when this triad occurs, then CLT is given by equation 3.14.

$$CLT = \frac{C_{(L_{T-1},L_T,L_{T+1})}}{N_{(L_{T-1},L_T,L_{T+1})}} \tag{3.14}$$

If in equation 3.14 we substitute the two quantities with the respective ones for triads of POS tags instead of lemmas, we get CPT. Finally, *Word FOC Test Set* ($WFTS$) is the frequency of the target word in the test set.

Three SVM classifiers are trained in order to decide whether or not to use the respective base WSD method for a specific term occurrence. Training is based on the terms' feature vectors, which have the following form: $(X_{i1}, ..., X_{iN}, C_i)$, where

[3]When L_T is at the beginning of a sentence, L_{T-1} is the pseudo-lemma *start_of_text*. Similarly, when L_T is at the end of a sentence, L_{T+1} is the pseudo-lemma *end_of_text*.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

X_{in} is the value of the n_{th} feature from table 3.7, for the training example i, and C_i is the binary class attribute on which the respective SVM learns whether to trust the current disambiguation method ($C_i = 1$) or not ($C_i = -1$). The C_i values represent the success or failure of the respective WSD method in disambiguating the specific training instance. In order to improve the stability of the SVMs and improve their performance, we normalize all features in $[0, 1]$. We experimented with linear, polynomial and RBF kernels on a small subset of the training set (5 randomly selected files - on average approximately 4600 term occurrences each) and we repeated the experiment 5 times, each time using 10-fold cross validation. In all cases, the three SVMs provided better accuracies when the RBF kernel was used. Consequently, we decided to use the RBF kernel, with the default *gamma* parameter of the SVMLight implementation [46].

We decided to retain the overall ensemble complexity low by applying a simple arbiter based voting mechanism. Additionally, we experimented with four basic ensemble mechanisms that have proven to boost WSD performance in the case of Naive Bayes classifiers [33]. All the ensemble mechanisms were applied after calibrating the SVM outputs into the value range of $[-1, 1]$. Additional experimental results, reported later, show that using a linear classifier (Maximum Entropy) as a top-level classifier in the ensemble, does not improve performance. The description of all tested ensemble mechanisms, including the proposed arbiter-based voting, follows.

Ensemble Mechanism 1 and 2: Simple voting (SV) and weighted voting of the positive class normalized SVM outputs. The simple voting is summarized in equation 3.15, where $SVMV_i(x)$ is the value (confidence score) of the ith support vector machine (corresponding to the ith base WSD approach) for instance x. The mechanism takes into account only the positive SVMV values. Consequently when all the i values are negative this means that the mechanism does not trust any of the WSD methods, and it cannot disambiguate the specific word occurrence. The weighted voting multiplies each $SVM_i(x)$ with the accuracy A_i of the respective WSD method (SAN, PR, FS) in the training set.

$$\arg\max_i SVMV_i(x) \tag{3.15}$$

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Ensemble Mechanisms 3 and 4: These two mechanisms are variations of 1 and 2 respectively, that consider all the SVMV values, both positive and negative. The difference is that when all the $SVMV_i(x)$ values are negative, the mechanism decides to trust the WSD method corresponding to the least negative value.

In all the aforementioned ensembles each SVM is autonomous and its confidence is examined independently of the other two. Note also that in the case of ensembles 1 and 2, when all SVM values are negative, the algorithm decides not to disambiguate the examined word occurrence, since there is no SVM with positive confidence.

Ensemble Mechanism 5: Arbiter-based voting counts in a pairwise manner the number of times each SVM had larger SVMV value. Let (k, l) be all possible pairs of the underlying SVMs and m another SVM acting as an arbiter. If $SVMV_k$,$SVMV_l$,$SVMV_m$ are the corresponding SVM values for word occurrence i, and A_k, A_l, A_m the accuracies of the underlying base WSD methods respectively (SAN, PR, FS) then we compute $SVMV'_k$ (and $SVMV'_l$ in the same manner) with the following rule: if $SVMV_m > SVMV_k$ then $SVMV'_k$ becomes as shown in equation 3.16.

$$SVMV'_k = SVMV_k + (SVMV_m - SVMV_k) \cdot \sqrt{(A_m - A_k)^2 + (SVMV_m - SVMV_k)^2} \tag{3.16}$$

Else, if $SVMV_m < SVMV_k$ then $SVMV'_k$ becomes as shown in equation 3.17.

$$SVMV'_k = SVMV_k - (SVMV_k - SVMV_m) \cdot \sqrt{(A_m - A_k)^2 + (SVMV_m - SVMV_k)^2} \tag{3.17}$$

Else, if $SVMV_m = SVMV_k$ then $SVMV'_k = SVMV_k$.

Since there is no global consensus on which of the three SVMs could act as arbiter, and there is no unbiased external arbiter, each one of the i SVMs acts as an arbiter for the remaining $i - 1$, and announces its preference. Eventually, the most voted SVM indicates the WSD method to use. The intuition behind this set up, implemented by formulas 3.16 and 3.17, is that each arbiter takes into account the general WSD performance of the two compared methods (denoted by the overall accuracy of the corresponding WSD method). It also considers the degree of disagreement between itself and each of the other two SVM (denoted by the differences in the respective

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Algorithm 3 SupervisedWSD(TS, T, U)

Require: A sense-annotated training set (TS), a part of speech tagged text (T), a word thesaurus (U).

Ensure: A mapping of the terms in T to thesaurus' senses. *Training(TS,U)*

1: **for all** terms $t \in TS$ **do**
2: SANSenses[t]=disambiguate(SAN,t)
3: PRSenses[t]=disambiguate(PR,t)
4: FSSenses[t]=disambiguate(FirstSense,t)
5: **end for**
6: SANModel = Train(SANSenses, TS)
7: PRModel = Train(PRSenses, TS)
8: FSModel = Train(FSSenses, TS)
Disambiguate(T,U,SANModel, PRModel, FSModel)
9: **for all** terms $t \in T$ **do**
10: WSDMethod = Predict(t, Ensemble(SANModel, PRModel, FSModel))
11: disambiguate(WSDMethod,t)
12: **end for**

SVMV values). Based on that, the arbiter recalculates the SVMV values of the compared methods and votes for the SVM with the highest SVMV value.

Algorithm and Complexity

The proposed WSD approach can be summarized as algorithm 3. The time cost of the training step consists of the time required to disambiguate all training terms with all the base methods and the time to train the three SVMs. In [121] we showed that the construction time of the semantic networks is $O(n \cdot k^{l+1})$ where n is the number of words we disambiguate, k is the maximum branching factor of the used thesaurus nodes, and l is the maximum semantic path length in the thesaurus. The execution of the SANs, costs $O(n^2 \cdot k^{2l+3})$. The execution of PR costs $O(n^2 \cdot k^{\frac{3}{2}l+3})$, in the worst case where the network has $n \cdot k^{\frac{l}{2}+1}$ nodes and $n \cdot k^{l+2}$ edges. The execution of FS costs $O(n)$. PR and SAN share the same networks which are constructed once, so the overall time cost for WSD of training terms is $O(n^2 \cdot k^{\frac{3}{2}l+3})$. For the SVMs training (SVM RBF kernel training) we used the quadratic optimizer of SVMLight [46]. The overall training time cost is one-time cost and the algorithm requires few training

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

#Training Files	5	10	20	40	60	80	130	ALL (186)
Avg. # Training Instances	4566	8500	16407	32612	48524	64683	106657	169958
Avg. # SAN SVM Support Vectors	51	80	188	322	416	667	1186	1498
Avg. # PR SVM Support Vectors	44	52	98	162	194	339	567	715
Avg. # FirstSense SVM Support Vectors	47	55	125	175	264	465	832	1062
Arbiter Voting, Senseval 2	40.04%	46.47%	46.7%	50.9%	65.02%	65.1%	65.1%	65.1%
Arbiter Voting, Senseval 3	46.76%	52.62%	55.7%	61.42%	63.9%	65.7%	65.7%	65.7%

Table 3.8: Average number of training instances and support vectors for each SVM.

examples to obtain its top performance.

The space complexity of the training step is $O(n^2 \cdot k^{2l+3})$. The disambiguation step is fast (at worst the execution time of SANs and at best, the execution time of FS) and does not require much memory, since, as reported in table 3.8, a few hundreds of support vectors in average need to be stored in each SVM model. Note also that only one WSD method (the most suitable according to the classifiers' ensemble) is executed for each test term.

Comparison with Related WSD methods

In this section we theoretically compare our method against four other state of the art supervised WSD methods. In the SenseLearner method [67] the authors suggest the construction of seven semantic models, which are trained using the Timbl memory based learning algorithm. The major drawback of this method is that the coverage of the target words in the disambiguation phase is limited to those words previously seen in the training corpus. In contrast, our method does not have this limitation; the only complication are the values of CWA and CLT, which are zero.

The Simil-Prime method [49] learns generic semantic classes, thus alleviating the aforementioned limitation of coverage. Then it casts back the finer grained senses from the generic semantic classes learned, using heuristical mapping. The major drawback of this method is the use of heuristics, which cannot guarantee that finer senses will not be missed. Another drawback is the fact that a decision-tree based implementation of the k-nn classifier is used. Though faster than the typical k-nn

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

classifier, the execution cost of disambiguation (mainly space complexity) is still high, since many training examples need to be reexamined for each target word. In contrast our method considers at most 3% of the training examples (table 2 - ratio of training instances used as support vectors).

The SSI method [77] uses the HITS algorithm as a means of discovering the dominant senses in a given word context $\sigma = w_1, w_2, ..., w_n$. A significant disadvantage of this approach is the use of HITS itself, which is prone to clique-attack: a small set of strongly interconnected senses can gain advantage against the senses of the semantic graph that are less interconnected. In contrast, the semantic network based methods employed in our approach use PageRank and constrained spreading of activation respectively, which solve the aforementioned problem.

Finally, in [40] the authors propose a memory-based learning approach, that uses voting among word-experts to decide on the correct sense. This memory-based method stores all instances in memory during training and testing, which results in high space and time complexity.

Experimental Evaluation

With regards to the evaluation of the training process, this is repeated for a varying number of training files to observe the performance of our system for a varying size of training instances. We randomly select (with a uniform distribution) 5,10,20,40,60,80 and 130 documents from the SemCor set, and perform training. We proceed by evaluating our method on the test sets. We used 10 iterations (random selections of training documents) for each training set size. We also performed training once, using the complete set of SemCor documents (186 files). Table 3.8 summarizes the training process and shows the average number of training instances and support vectors for each SVM. It also presents the accuracy of our best set up, which is the use of the arbiter-based voting (ensemble mechanism 5), in the Senseval 2 and 3 data sets. By consulting table 3.8 we conclude that using few training instances (less than 40% of SemCor) the proposed ensemble learns to trust the correct WSD method per case and achieves the highest possible accuracy.

Next, we compare with the methods of [67] (SenseLearner), [49] (Simil-Prime),

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Method	SenseLearner	Simil-Prime	SSI	WE	EM1	EM2	EM3	EM4	Arbiter Voting	FS	U-Bound
Senseval 2	66.22	66.4	N/A	63.2	63.8	63.8	63.8	63.8	**65.1**	63.7	80.7
Senseval 3	63.28	66.1	60.4	N/A	64.2	64.5	64.1	64.3	**65.7**	61.3	76.6

Table 3.9: Accuracies (%) on Senseval 2 and 3 All English Words Data Sets.

[77] (SSI), [40] (WE), on the Senseval 2 and 3 data sets. Table 3.9 shows the respective accuracies, where available. We also report the performance of the other tested ensemble mechanisms, as well as the FS baseline. We can compare with the method of Brody et al. [17] only in the noun POS of Senseval 3 data set, since their method's evaluation is limited to that. Arbiter voting achieves an accuracy of 74.3% in the Senseval 2 nouns and an accuracy of 74.18% in the Senseval 3 nouns. Brody et al. report an accuracy of 63.9% in Senseval 3 nouns (Senseval 2 is N/A) with an upper bound lower than 70%. In all, the proposed system ranks among the top 3 approaches from all the compared systems in table 3, which provide the best ever reported results in Senseval 2 and 3 *English all words* task. The achieved performance requires less that 40% of SemCor for training.

In order to test the effect of using SVMs in the classifier level, we conducted experiments using Maximum Entropy (ME) classifiers instead. The results were worse than using SVM: in Senseval 2 we obtained an overall accuracy of 63.95% and in Senseval 3 63.85%. We also experimented using ME as the ensemble mechanism at the decision (top) level. For this, we partitioned SemCor instances into two sets, one for training the SVMs and another for training the ensemble mechanism. Despite the additional cost of training at the decision level, the results were worst than using the arbiter-based voting (60.9% in Senseval 3 and 65.5% in Senseval 2).

To analyze whether all three WSD methods are necessary or not, we consider an unerring "oracle", an ideal decision level mechanism, that would always select the correct method among the three dictionary-based. Initially, the upper-bound of accuracy is 80.76% in Senseval 2 and 76.65% in Senseval 3, when all three methods are used. They fall to 76.03% and 73.06% respectively when the PR method is removed from the base level, and to 77.72% and 71.85% respectively, when the SAN method is removed. Finally, by removing FS, the upper bounds drop to 69.73% and 63.36%

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Collection	Compactness (Nouns only)	SAN	PR	FS	Ensemble	UB
Senseval 2	48.2	49.3	57.9	61.3	65.1	80.7
Senseval 3	45.4	47.4	51.7	63.4	65.7	76.6

Table 3.10: Synopsis of WSD Results in Senseval 2 and 3.

respectively. This shows that all three WSD methods are necessary and cannot be omitted. A fourth WSD method should ideally disambiguate correctly all the terms missed by the other three methods and it should favor the less frequent senses (i.e., senses 3-4 and below in WordNet).

3.5 Discussion of Experimental Results

In this chapter, we have presented four new knowledge-based methods for WSD based on WordNet (Compactness, SANs, PR and an ensemble WSD system). A synopsis of the results in the two Senseval competitions is shown in table 3.10. The table shows the accuracies for all methods in full coverage. Note however that the accuracy of the Compactness-based method refers only to nouns, since the method cannot disambiguate other POS. As shown, SAN and PR provide state of art results in knowledge-based WSD, with overall accuracies of approximately 55%, without any type of training, and with average polysemy ranging from 5.3 to 7.2. Both methods are outperformed by the FS method, which is the supervised baseline. The ensemble method provides state of the art performance in supervised WSD, surpassing the FS with statistical significance at the 0.95 confidence level. UB is the upper bound that the ensemble can reach, if a learner fits the data more, but that could reside in an overfit.

An argument can be raised regarding the fact that the reported accuracies may seem low and close to the FS baseline. It is true, though, that the best current state of the art methods all achieve small improvements $(1 - 3\%)$ compared to FS. This is not trivial, if one considers the fact that the human interannotator agreement is approximately 70% and, thus, accuracies of 65% essentially approximate the human

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

performance in WSD.

In all, in this chapter we have introduced WSD methods that are among the top-performing methods in the current WSD bibliography [79]. Our top-performing method is a multilayered WSD system based on an ensemble of three WordNet-based WSD algorithms. The method trains a set of SVM classifiers, one for each WSD algorithm, and learns which WSD method to trust depending on a feature vector of the target term. The main advantages of this method are: (1) state of the art accuracy in unrestricted text, (2) limited training requirements to achieve top performance, (3) low space complexity, since the classifiers are trained on a very small number of features and the stored support vectors are on average 3% of the training instances, and (4) the disambiguation step of the algorithm has lower time complexity for the disambiguation than existing supervised WSD methods. Overall, the proposed approach competes well (in terms of accuracy) against state of the art methods for unrestricted text WSD.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Chapter 4

Omiotis: A Thesaurus-based Measure of Semantic Relatedness

Relatedness between texts can be perceived in several different ways. Primarily, one can think of surface (string) relatedness or similarity between texts, which can be easily captured by a vectorial representation of texts and a standard similarity measure (e.g., cosine, Dice, Jaccard etc.). Such models have had high impact in information retrieval over the past decades. Several improvements have been proposed for such techniques, towards inventing more sophisticated weighting schemes for the text terms (e.g. TF-IDF and its variations). Other directions explore the need to capture the latent semantic relations between dimensions (words) in the constructed vector space model, by using techniques of latent semantic analysis [52]. Another aspect of text relatedness, probably of equal importance, is the semantic relatedness between two text segments. For example, the sentences "*The shares of the company dropped 14 cents*" and "*The business institution's stock slumped 14 cents*" have an obvious semantic relatedness, as explained in the introduction, which traditional measures of text similarity fail to recognize. In this chapter we propose Omiotis, a measure of relatedness between texts, which takes into account both the surface and the semantic relatedness of words, performs better than the traditional surface (string) matching models, and can handle cases like the above.[1]

[1] *Omiotis* is the Greek word for relatedness or similarity.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

The measure is based on the semantic representation of the WSD methods introduced in the previous chapter. The core of the measure is SR, a new measure of semantic relatedness between senses, that is extended to measure semantic relatedness between words. Omiotis constitutes the final extension to measure semantic relatedness between text segments. The word relatedness measure is based on the construction of semantic links between individual words, according to a word thesaurus (in our case, WordNet). Each pair of words is potentially connected via one or more semantic paths, each one comprising one or more edges that connect intermediate thesaurus nodes. To weigh the semantic path we consider three key factors: (a) the length of the semantic path; (b) the intermediate nodes' specificity, denoted by the node depth in the thesaurus' hierarchy; and (c) the types of the semantic edges that compose the path. The three factors allow our measure to perform well in complex linguistic tasks, that require more than simple similarity, such as the SAT Analogy Test (section 5.1).[2] To the best of our knowledge, SR and Omiotis are the first measures of semantic relatedness that consider in tandem all three factors. Omiotis integrates semantic relatedness at the word level with statistical information about words at the text level, and provides a semantic relatedness measure between texts.

The contributions of this chapter are: 1) a new measure for computing semantic relatedness between words, which exploits all of the semantic information a thesaurus can offer, including semantic relations crossing parts of speech (POS), while taking into account relation weights and the depth of the thesaurus' nodes; 2) a new measure for computing semantic relatedness between texts, Omiotis, that does not require any type of training; 3) thorough experimental evaluation on benchmark data sets to measure the performance of word-to-word similarity in word analogy, as well as experiment on three text related tasks (paraphrase recognition, document similarity detection, document classification) to evaluate the performance of the text-to-text relatedness measure. An additional practical contribution of the thesis is a publicly available system [120] that can be used to obtain a pre-computed semantic relatedness score between any pair of WordNet senses, which facilitates incorporating our

[2] http://www.aclweb.org/aclwiki/index.php?title=SAT_Analogy_Questions

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

semantic relatedness measure in many retrieval tasks.[3]

The key features of the proposed measures are: (a) it constructs semantic links between all word senses in WordNet and pre-computes a relatedness score between every pair of WordNet senses; (b) it computes the semantic relatedness for a pair of words by taking into account the relatedness of their corresponding WordNet senses; (c) it computes a semantic relatedness score for any two given text segments using the relatedness at word-level. Depending on the task, the computation of semantic relatedness can be modified to take into account all or some of the senses of each word, or all or some of the words in each text, depending on the word importance or sense importance in context. This allows Omiotis to be adapted in various text related tasks, without modifying the main process of computing relatedness. In the section 4.1 below, we formally define our semantic relatedness measure for senses. In section 4.2 we present its extension to measure relatedness between words, and in section 4.3 we provide a detailed justification of our design decisions. In section 4.4 we define Omiotis, and in section 5.1 we explain how Omiotis can be embedded in several applications.

4.1 Semantic Relatedness Between a Pair of Concepts

In order construct semantic paths between words, we reuse the idea of constructing semantic networks connecting words, that we presented in section 3.3.1 [121].

Figure 4.1 gives an example of the construction of a semantic network for two words t_i and t_j. For simplicity, we assume the construction of a semantic path between the highlighted senses $S.i.2$ and $S.j.1$ only (Initial Phase), though we could do the same for every possible pair of two words' senses. Initially, the two sense nodes are expanded using all the semantic links offered by WordNet. The semantic links of the highlighted senses, as found in the thesaurus, are added as edges and the senses they point to are added to the network as nodes (Network Expansion). The expansion

[3]Available at `http://omiotis.hua.gr/WebSite/`

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

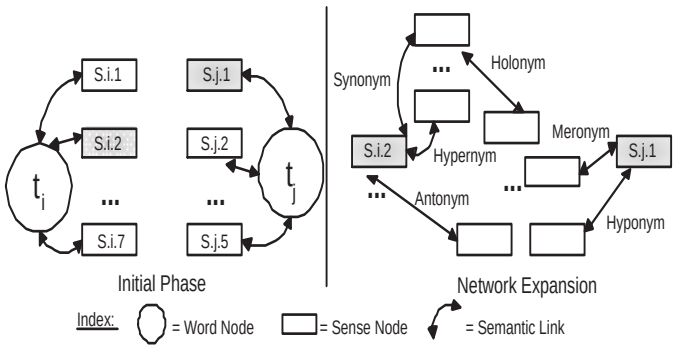

Figure 4.1: Constructing semantic networks from word thesauri.

process is repeated recursively until the shortest path between $S.i.2$ and $S.j.1$ is found; a more detailed description of this process is provided as algorithm 4. If no path is found from $S.i.2$ to $S.j.1$, then the senses and consequently the words are not semantically related.

The semantic relatedness of a pair of concepts is measured over the constructed semantic network. The measure considers the path length, captured by *semantic compactness*, and the path depth, captured by *semantic path elaboration*, which are defined in the following. Note that in the previous chapter we introduced a measure for WSD based on the idea of *compactness* that we initially proposed in [63]. That measure used only nouns and the hypernym relation. In this chapter, the measure is extended to exploit all of WordNet's relations and the noun, verb, and adjective parts of speech. We also define a new *compactness* measure (definition 3) as on of the core components of the Omiotis measure.

Definition 3 *Given a word thesaurus O, a weighting scheme that assigns a weight $e \in (0, 1)$ to each edge type (each edge is assigned the weight of its type), a pair of senses $S = (s_1, s_2)$, and a path of length l connecting the two senses, the semantic compactness of S ($SCM(S, O)$) is defined as: $SCM(S, O) = \prod_{i=1}^{l} e_i$, where $e_1, e_2, ..., e_l$ are the weights of the path's edges. If $s_1 = s_2$ then $SCM(S, O) = 1$. If there is no path between s_1 and s_2 then $SCM(S, O) = 0$.*

Note that *compactness* takes the path length into account and is bound in [0, 1].

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

WordNet 2.0 Edge Type	Probability of Occurrence
hypernym/hyponym	0.61
nominalization	0.147
category domain	0.094
part meronym/holonym	0.0367
region domain	0.0238
similar	0.02
usage domain	0.016
member meronym/holonym	0.014
antonym	0.0105
verb group	0.01
also see	0.0091
attribute	0.00414
entailment	0.00195
cause	0.00158
substance meronym/holonym	0.00089
derived	0.0003
participle of	$3.4E-06$

Table 4.1: Probability of occurrence for every edge type in WordNet 2.0.

Higher *compactness* between senses declares higher semantic relatedness. The intuition behind the weighting of the edge types is that certain types provide stronger (more straightforward, e.g., hypernym/hyponym edges) semantic connections than others. Considering that human editors of WordNet tend to use the stronger relation types more often than weaker ones, a straightforward solution is to define the weights of the edge types proportionally to their frequency of occurrence in WordNet 2.0. The weights assigned to each type using this solution are shown in table 4.1 and are in accordance to those found in [109]. The table shows the probability of occurrence in WordNet 2.0 of every possible edge type in decreasing order of probability.

The depth of nodes that belong to the path also affects term relatedness. A standard means of measuring depth in a word thesaurus is the hypernym/hyponym hierarchical relation for the noun and adjective POS and hypernym/troponym for the verb POS. For the adverb POS the related *stem adjective* sense can be used to measure its depth. A path with shallow sense nodes is more general compared to

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

a path with deep nodes. This parameter of semantic relatedness between terms is captured by the measure of *semantic path elaboration* and is introduced in definition 4.

Definition 4 *Given a word thesaurus O , a pair of senses $S = (s_1, s_2)$, where s_1,$s_2 \in O$ and $s1 \neq s2$, and a path between the two senses of length l, the semantic path elaboration of the path (SPE(S,O)) is defined as:*

$$SPE(S,O) = \prod_{i=1}^{l} \frac{2d_i d_{i+1}}{d_i + d_{i+1}} \cdot \frac{1}{d_{max}},$$

where d_i is the depth of sense s_i according to O, and d_{max} the maximum depth of O. If $s_1 = s_2$ and $d = d_1 = d_2$ then $SPE(S,O) = \frac{d}{d_{max}}$. If there is no path from s_1 to s_2 then $SPE(S,O) = 0$.

A path of length l comprises l+1 nodes, thus when $i = l$, d_{i+1} is the last node in the path. Essentially, each factor of SPE is the harmonic mean of the depths of two adjacent senses along the path, normalized to the maximum thesaurus depth. The harmonic mean is preferred over the average of depths, since it offers a lower upper bound. *Compactness* and *Semantic Path Elaboration* capture the two most important parameters of measuring semantic relatedness between terms [18], namely path length and senses depth in the used thesaurus. We combine these two measures as follows in the definition of the *Semantic Relatedness* between two terms (definition 5).

Definition 5 *Given a word thesaurus O, and a pair of senses $S = (s_1, s_2)$ the semantic relatedness of S (SR(S,O)) is defined as $max\{SCM(S,O) \cdot SPE(S,O)\}$, over all the paths that connect s_1 and s_2.*

Given a word thesaurus, there can be more than one semantic paths connecting two senses. The senses' *compactness* can take different values for different paths. In these cases, we use the path that maximizes the semantic relatedness. For its computation we introduce algorithm 4, which is a modification of Dijkstra's algorithm for finding the shortest path between two nodes in a weighted directed graph. The modification made is to incorporate the multiplication of edges weights, substituting the sum in

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Algorithm 4 Maximum-Semantic-Relatedness(G,u,v,w)

Require: A directed weighted graph G, two nodes u, v and a weighting scheme $w : E \rightarrow (0..1)$, where E is the set of all edge types.

Ensure: The path from u to v with the maximum product of the edges weights.

Initialize-Single-Source(G,u)

1: **for all** vertices $v \in V[G]$ **do**
2: $d[v] = -\infty$
3: $\pi[v] = NULL$
4: **end for**
5: $d[u] = 1$

Relax(u, v, w)

6: **if** $d[v] < d[u] \cdot w(u, v)$ **then**
7: $d[v] = d[u] \cdot w(u, v)$
8: $\pi[v] = u$
9: **end if**

Maximum-Relatedness(G,u,v,w)

10: Initialize-Single-Source(G,u)
11: $S = \emptyset$
12: $Q = V[G]$
13: **while** $v \in Q$ **do**
14: $s =$ Extract from Q the vertex with the maximum d
15: $S = S \cup s$
16: **for all** vertices $k \in$ Adjacency List of s **do**
17: Relax(s,k,w)
18: **end for**
19: **end while**
20: return the path following all the ancestors π of v back to u

the original algorithm. The proof of the algorithm's correctness follows in the next theorem, and it is based on the respective proof of the Dijkstra algorithm in [23]. In algorithm 4, d holds an estimate of the weight of the shortest path from the source to each node in the graph, and π holds the predecessor of each node (so that at the end we know the exact shortest path from the source to every node in the graph).

Theorem 1 *Given a word thesaurus O, an edges weighting function $w : E \rightarrow (0, 1)$, where a higher value declares a stronger edge, and a pair of senses $S(s_s, s_f)$ declaring source (s_s) and destination (s_f) vertices, then the $SCM(S, O) \cdot SPE(S, O)$ is maximized for the path returned by Algorithm 4, by using the weighting scheme*

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

$e_{ij} = w_{ij} \cdot \frac{2 \cdot d_i \cdot d_j}{d_{max} \cdot (d_i + d_j)}$, *where* e_{ij} *the new weight of the edge connecting senses* s_i *and* s_j.

Proof 1 *We will show that for each vertex* $s_f \in V$, $d[s_f]$ *is the maximum product of edges' weight through the selected path, starting from* s_s, *at the time when* s_f *is inserted into* S. *From now on, the notation* $\delta(s_s, s_f)$ *will represent this product. Path* p *connects a vertex in* S, *namely* s_s, *to a vertex in* $V - S$, *namely* s_f. *Consider the first vertex* s_y *along* p *such that* $s_y \in V - S$ *and let* s_x *be* y*'s predecessor. Now, path* p *can be decomposed as* $s_s \rightarrow s_x \rightarrow s_y \rightarrow s_f$. *We claim that* $d[s_y] = \delta(s_s, s_y)$ *when* s_f *is inserted into* S. *Observe that* $s_x \in S$. *Then, because* s_f *is chosen as the first vertex for which* $d[s_f] \neq \delta(s_s, s_f)$ *when it is inserted into* S, *we had* $d[s_x] = \delta(s_s, s_x)$ *when* s_x *was inserted into* S.

Because s_y *occurs before* s_f *on the path from* s_s *to* s_f *and all edge weights are nonnegative and in* $(0, 1)$ *we have* $\delta(s_s, s_y) \geq \delta(s_s, s_f)$, *and thus* $d[s_y] = \delta(s_s, s_y) \geq \delta(s_s, s_f) \geq d[s_f]$. *But both* s_y *and* s_f *were in* $V - S$ *when* s_f *was chosen, so we have* $d[s_f] \geq d[s_y]$. *Thus,* $d[s_y] = \delta(s_s, s_y) = \delta(s_s, s_f) = d[s_f]$. *Consequently,* $d[s_f] = \delta(s_s, s_f)$ *which contradicts our choice of* s_f. *We conclude that at the time each vertex* s_f *is inserted into* S, $d[s_f] = \delta(s_s, s_f)$.

Next, to prove that the returned maximum product is the $SCM(S,O) \cdot SPE(S,O)$, *let the path between* s_s *and* s_f *with the maximum edge weight product have* k *edges. Then, Algorithm 1 returns the maximum* $\prod_{i=1}^{k} e_{i(i+1)} = w_{s2} \cdot \frac{2 \cdot d_s \cdot d_2}{d_{max} \cdot (d_s + d_2)} \cdot w_{23} \cdot \frac{2 \cdot d_2 \cdot d_3}{d_{max} \cdot (d_2 + d_3)} \cdot \ldots \cdot w_{kf} \cdot \frac{2 \cdot d_k \cdot d_f}{d_{max} \cdot (d_k + d_f)} = \prod_{i=1}^{k} w_{i(i+1)} \cdot \prod_{i=1}^{k} \frac{2 d_i d_{i+1}}{d_i + d_{i+1}} \cdot \frac{1}{d_{max}} = SCM(S,O) \cdot SPE(S,O)$.

4.2 Semantic Relatedness Between a Pair of Words

Based on definition 5, which measures the semantic relatedness between a pair of senses S, we can define the semantic relatedness between a pair of terms $T(t_1, t_2)$ as in definition 6.

Definition 6 *Let* O *be a word thesaurus* O, *let* $T = (t_1, t_2)$ *be a pair of terms for which there are entries in* O, S_1 *be the set of senses of* t_1 *and* S_2 *the set of senses of* t_2 *in* O. *If* S_k, $k = 1..|S_1| \cdot |S_2|$ *are all the possible sets of senses pairs* (s_i, s_j), *with*

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

$s_i \in S_1$ and $s_j \in S_2$, then the semantic relatedness of T ($SR(T, S_k, O)$) is defined as $max\{SCM(S_j, O) \cdot SPE(S_j, O)\}$, for all $j = 1..|S_1| \cdot |S_2|$. The semantic relatedness between two terms t_1, t_2 where $t_1 \equiv t_2 \equiv t$ and $t \notin O$ is defined as 1. The semantic relatedness between t_1, t_2 when $t_1 \in O$ and $t_2 \notin O$, or vice versa, is considered 0.

In the remainder of this thesis, the $SR(T, S_k, O)$ for a pair of terms will be denoted as $SR(T)$, to ease readability. Note also that this measure of semantic relatedness between terms, inherently performs WSD, if one term is taken to be the target one and the other its (single-word) context: For a given word t_1, the semantic relatedness to another word t_2 might be maximized with the sense s_{11} of t_1, and for a new given word t_3, it might be maximized for another sense, e.g. s_{12}. In [119] we present the findings of using $SR(T)$ as a WSD measure; the performance was similar to SANs, on Senseval 2 and 3 data sets.

4.3 Analysis of the SR Measure

In this section we present the rationale behind the definitions 2, 4 and 5, by providing theoretical and/or experimental evidence for the decisions made on the design of the measure. We illustrate the advantages and disadvantages of the different alternatives using simple examples and argue for our decisions. Finally, we discuss the advantages of SR against previous measures of semantic relatedness and its possible caveats.

The list of decisions made for the design of our semantic relatedness measure comprises: a) use of senses in all POS, instead of noun senses only, b) use of all semantic edge types found in WordNet, instead of the IS-A relation only, c) use of edge weights, and d) use of senses' depth as a scaling factor. It is important to mention that measures of semantic relatedness differ from measures of semantic similarity, which traditionally use hierarchical relations only and ignore all other types of semantic relations. In addition, both concepts differ from semantic distance, in the sense that the latter is a metric.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Use all POS Information

Firstly, we shall argue that the use of all POS in designing a semantic relatedness measure is important, and can increase the coverage of such a measure. The rationale supporting this decision is fairly simple. Current data sets for evaluating semantic relatedness or even semantic similarity measures are restricted to the noun POS (Rubenstein and Goodenough's 65 word pairs [96], Miller and Charles' 30 word pairs [72], the Word-Similarity-353 collection [32]) and, thus, cannot pinpoint the caveat of omitting the remaining parts of speech. However, text similarity tasks and their benchmark data sets comprise more than nouns. Throughout the following analysis, the reader should keep in mind that the resulting measure of semantic relatedness between words is destined to be embedded in a text-to-text semantic relatedness measure, as discussed later.

The following two sentences are a paraphrase example taken from the Microsoft Paraphrase Corpus [29] and show the importance of using other POS, apart from nouns, such as verbs:

> "The **charges** of **espionage** and **aiding** the **enemy** can **carry** the **death penalty**."
>
> "If **convicted** of the **spying charges** he could **face** the **death penalty**."

Words that appear in WordNet 2.0 are written in bold and stopwords have been omitted for simplicity.[4] The two sentences have many nouns in common (charges, death, penalty), but there are also pairs of words across the two sentences that can provide evidence that the two sentences are paraphrases of each other. For example **espionage** and **spying** have an obvious semantic relatedness, as well as **enemy** and **spying**. Also, **charges** and **convicted**, as well as **penalty** and **convicted**. This type of evidence would have been disregarded by any measure of semantic relatedness or similarity that uses only the noun POS and WordNet's hierarchy. Examples of such measures are: the measure of Sussna [114], Wu and Palmer [135], Jiang and

[4]The stopwords' list that we used is available at `http://www.db-net.aueb.gr/gbt/resources/stopwords.txt`

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

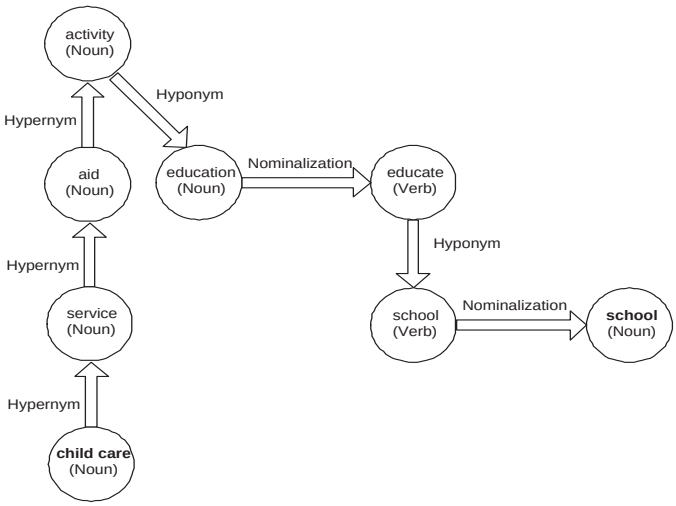

Figure 4.2: Semantic path from **child care** to **school**.

Conrath [45], Resnik [91, 92], and the WordNet-based component of Finkelstein et al. [32]. From this point of view, the decision to use all POS information expands the potential matches found by the measure and allows using the measure in more complicated tasks, like paraphrase recognition, text retrieval, and text classification.

Use Every Type of Semantic Relations

The decision to use all parts of speech in the construction of the semantic graphs requires using all semantic relations instead of merely taxonomic (IS-A) ones. Moreover, this decision was based on evidence from related literature. The work of Richardson et Al. [107] provides experimental evidence that measuring semantic similarity by incorporating non-hierarchical link types (i.e., part meronym/holonym, member meronym/holonym, substance meronym/holonym) improves significantly the performance of such a measure; the experiments conducted by adopting a small variation of the Resnik's measure [91].

Hirst and St-Onge [39] reported that they discovered several limitations and missing connections in WordNet's relations during the construction of lexical chains from

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

sentences for the detection and correction of malapropisms.[5] They provided the following example using the pair of words in bold to report this caveat:

> "School administrators say these same taxpayers expect the schools to provide **child care** and **school** lunches, to integrate immigrants into the community, to offer special classes for adult students,."

The intrinsic connection between the nouns **child care** and **school**, which both exist in WordNet, cannot be discovered by considering only hierarchical edge types. This connection is depicted in figure 4.2, which shows the path in WordNet. By using all WordNet's relations, our measure is able to detect such connections and address problems of the aforementioned type.

Use Weights on Edges

Resnik [92] reports that simple edge counting, which implicitly assumes that links in the taxonomy represent equal distances, is problematic and is not the best semantic distance measure for WordNet. In a similar direction lie the findings of Sussna [114], who performed thorough experimental evaluation by varying edge weights in order to measure semantic distance between concepts. Sussna's findings, revealed that weights on semantic edges are a non-negligible factor in the application of their measure to WSD, and that their best results were reported when a weighting scheme for edges was used, in contrast to assigning each edge the same weight. For all these reasons, we decided to assign a weight to every edge type, and we chose the simple probability of occurrence for each edge type in WordNet, as our edge weighting scheme (see table 4.1). This very important factor is absent from several similarity measures proposed in the past, such as the measures of Leacock and Chodorow [55], Jarmasz and Szpakowicz [44], and Banerjee and Pedersen [6], which are outperformed in experimental evaluation by our measure.

[5]Malapropism is the bad use of a word due to confusion with another word having the same sound

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

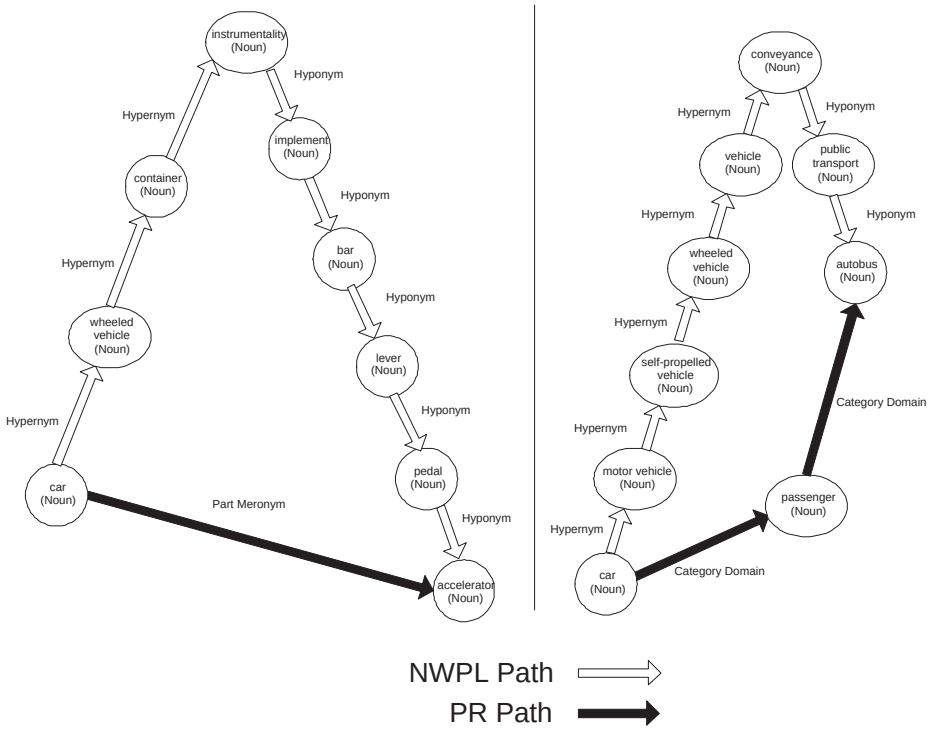

Figure 4.3: PR and NWPL paths for pairs: *car* and *accelerator* (left), *car* and *autobus* (right).

Use Depth Scaling Factor

Our decision to incorporate the depth scaling factor (SPE in definition 4) in the edge weighting mechanism was inspired by the thorough experimental evaluation conducted by Sussna [114], which provided evidence on the importance of the edge weighting factor in measures based on semantic networks. Our experiments on the Miller and Charles data set, show that the Spearman correlation with human judgements is much lower (7 percentage points) without the depth scaling factor, compared to the correlation when adopting the SPE factor (see definition 2).

Justification of SR Definitions

According to definition 2, the semantic compactness of a pair of concepts is the product of the depth-scaled weights of the edges connecting the two concepts. The

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

use of a product instead of a sum or normalized sum of edge weights is explained in the following.

Since there might be several paths connecting the two concepts, definition 5 selects the path that maximizes the product of semantic compactness (SC) and semantic path elaboration (SPE). For simplicity, we ignore the effect of the depth scaling factor (SPE in definition 4) and consequently, our aim is to find the path that maximizes $\prod_{i=1}^{l} e_i$, where $e_1, e_2, ..., e_l$ are the (non depth-scaled) weights of the edges in the path connecting two given concepts. Let us call this less elaborate version of our semantic relatedness measure *product relatedness* (PR). An alternative would be to define semantic compactness as the average of the weights in the path, i.e., $\frac{\sum_{i=1}^{l} e_i}{l}$. In this case, the semantic relatedness would be measured on the path that maximizes the latter formula. Let us name this alternative after *normalized weighted path length* (NWPL).

In the example of figure 4.3, we show how PR and NWPL compute the semantic relatedness for the term pair *car* and *accelerator* (left), and *car* and *autobus* (right). The path that maximizes the respective formulas of PR and NWPL using algorithm 4 and the edge weights in table 4.1, is also illustrated in the figure. For the pair *car* and *accelerator* the sum-based formula, normalized by the path length, selects a very large path in this example, with a final computed relatedness of 0.61, which is the weight of the hypernym/hyponym edges. PR finds that the path maximizing the product is the immediate part meronym relation from *car* to *accelerator*, with a computed relatedness of 0.0367, which is the weight of the part meronym edges. The main problem arising with NWPL is that it cannot distinguish among the relatedness between any pair of concepts in the hypernym/hyponym hierarchy of WordNet. In this example, NWPL computes the same relatedness (0.61) between every possible concept pair shown in the top figure. In contrast, PR is able to distinguish most of these pairs in terms of relatedness. More precisely, this behavior of PR is due to the fact that it embeds the notion of the path length, since the computed relatedness decays by a factor in the range $(0, 1)$ for every hop made following any type of semantic relation. Another example, that also shows the importance of considering all WordNet relations, is the one shown in the right part of figure 4.3, where NWPL and PR paths

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

have been computed for the term pair *car* and *autobus*. Again, NWPL selects a very large path, and does stays within the hypernym/hyponym tree. Note however, that the example on the right part of the figure points out the importance of the depth scaling factor as well, since the shown path for PR is considered if the depth scaling factor is used, otherwise it would follow the same path as NWPL.

Overall, NWPL would rather traverse through a huge path of hypernym/hyponym edges, than follow any other less important edge type which would decrease its average path importance. This behavior creates serious drawbacks: (a) lack of ability to distinguish relatedness among any pair of concepts in the same hierarchy (e.g., instrumentality and container are as related as instrumentality and accelerator), and (b) large increase of the actual computational cost of algorithm 4, due to the fact that NWPL will tend not to deviate from the hypernym/hyponym hierarchy, even if there is a direct semantic edge (other than hypernym/hyponym) connecting the two concepts, as shown in figure 4.3. Furthermore, by conducting experiments with NWPL in the 30 word pairs of Miller and Charles [72], we discovered that in almost 40% of the cases, NWPL produces the same value of semantic relatedness, equal to 0.61, being unable to distinguish them and creating many ties, while PR produces a different value from almost every pair. Thus, PR is a better option to use in our measure, as the semantic compactness factor.

Note that our measure is solely based on the use of WordNet, unlike measures of semantic relatedness that use large corpora, such as Wikipedia. Although such measures (e.g., Gabrilovich and Markovitch [34], and Ponzetto and Strube [88]) provide a larger coverage, including concepts that do not reside in WordNet, they require extensive training. Experimental evaluation in chapter 5, shows that our measure outperforms all the aforementioned word-to-word relatedness measures in all three data sets used. In the following section, we introduce Omiotis, the extension of SR for measuring text-to-text relatedness.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

4.4 Omiotis

The definition of semantic relatedness between a pair of terms can be extended to capture semantic relatedness between texts. Given a pair of texts A, B, for every word a_i in A we seek the word b_j of B that maximizes the semantic relatedness between a_i and b_j, according to definition 6. Besides the semantic relatedness between a_i and b_j we also consider the importance of each word using their TF*IDF values. We define λ_{a_i,b_j} of two words a_i and b_i as the harmonic mean of their TF*IDF values.

$$\lambda_{a_i,b_j} = \frac{2TF\_IDF(a_i)TF\_IDF(b_j)}{TF\_IDF(a_i) + TF\_IDF(b_j)} \tag{4.1}$$

Since we want to combine the importance of terms a_i and b_j, according to their TF*IDF values, with their semantic relatedness, eventually, for every word a_i of document A we seek the word $x(a_i)$ from document B for which:

$$x(a_i) = \arg\max_{j\in[1,|B|]}(\lambda_{a_i,b_j} \cdot SR(a_i, b_j)) \tag{4.2}$$

Similarly, we do the same for document B, seeking for every word b_j, the word $y(b_j)$ of document A, for which:

$$y(b_j) = \arg\max_{i\in[1,|A|]}(\lambda_{a_i,b_j} \cdot SR(a_i, b_j)) \tag{4.3}$$

We aggregate these scores for both directions (e.g., from document A to document B and vice versa) as shown in equations 4.4 and 4.5 respectively. Note that both directions are needed, to cover the cases where the number of words are not equal in the two texts. Finally, OMOIOTIS between A, B can be obtained by the formula shown in equation 4.6.

$$\zeta_1(A, B) = \frac{1}{|A|}\left(\sum_{i=1}^{|A|} \lambda_{a_i,x(a_i)} \cdot SR(a_i, b_{x(a_i)})\right) \tag{4.4}$$

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

$$\zeta_2(A,B) = \frac{1}{|B|}\left(\sum_{j=1}^{|B|} \lambda_{y(b_j),b_j} \cdot SR(a_{y(b_j)}, b_j)\right) \tag{4.5}$$

$$Omiotis(A,B) = \frac{1}{2}\left[\zeta_1(A,B) + \zeta_2(A,B)\right] \tag{4.6}$$

Note that measure $Omiotis(A,B)$ between texts A and B takes into account both the importance of the terms in the document (TF-IDF values), as well as the semantic relatedness of the terms across the two documents. In fact, the new measure extends the THESUS measure that we presented in [129], which could only handle noun words, and did not consider TF-IDF scores. Appendix B analyzes the complexity of Omiotis and presents a novel implementation of our measure, along with an on-line demo.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Chapter 5

Applications and Experimental Evaluation

5.1 Applications of Semantic Relatedness

In this section we describe the methodology of incorporating semantic relatedness between pairs of words or pairs of text segments into well-known text related tasks.

Word Similarity

In 1965, Rubenstein and Goodenough [96] obtained synonymy judgements from 51 human subjects on 65 pairs of words, in an effort to investigate the relationship between similarity of context and similarity of meaning (synonymy). Since then, the idea of evaluating computational measures of semantic relatedness by comparing against human judgments on a given set of word pairs has been widely used, and even more data sets have been developed. The proposed measure of semantic relatedness between words (SR), introduced in definition 6, can be used directly in such a task, in order to evaluate directly the basis of the Omiotis measure, which is the measurement of word-to-word semantic relatedness. The application is straightforward: let n be all the pairs of words in the word similarity data set used; then, the semantic relatedness for every pair is computed, through the use of $SR(T, S, O)$, as defined

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

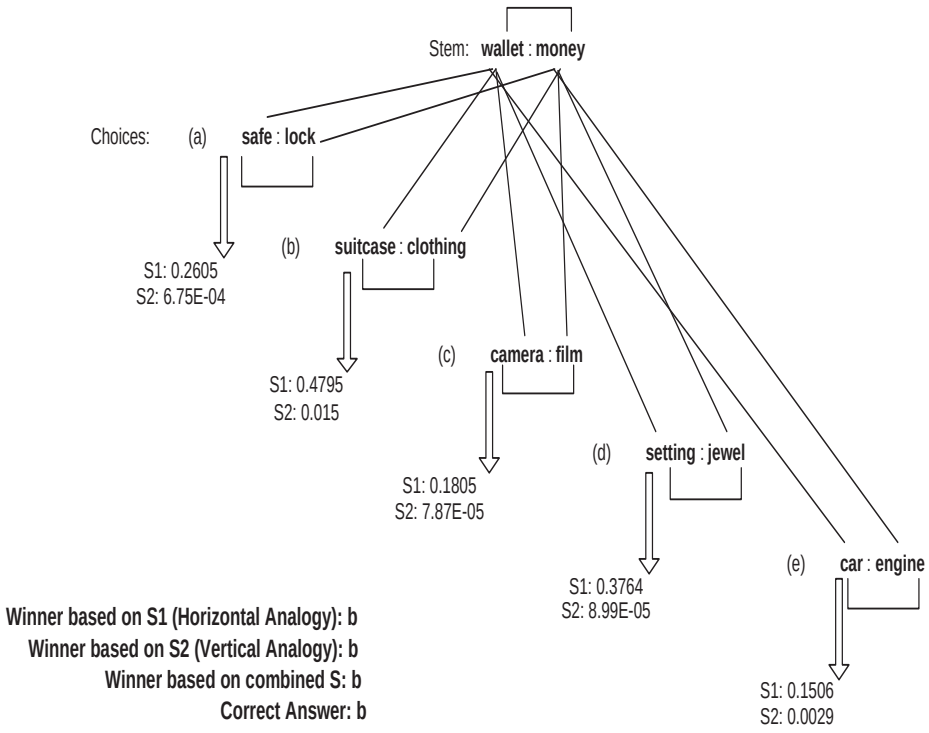

Figure 5.1: Example of computing SR in a given SAT question.

in 6; the computed values are sorted in decreasing order, and the produced ranking of similarities is compared against the "gold standard" ranking of humans, using Spearman correlation. Additional measures of semantic relatedness can be compared against each other by examining the produced values of the Spearman correlation.

SAT Analogy Tests

The problem of identifying similarities in word analogies among pairs of words is a difficult problem and it has been standardized as a test for assessing the human ability for language understanding in the well known SAT analogy tests (Scholastic Aptitude Tests). SAT tests in general are used as admission tests by secondary schools in the United States. The aim is to locate out of five pairs of words the one that presents the most similar analogy to a target pair.

Although it is difficult for machines to model the human cognition of word analogy, several approaches exist in the bibliography that attempt to tackle this problem.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Previous approaches can be broadly categorized in corpus-based, lexicon-based, and hybrid. Some examples of corpus-based approaches are those of Turney [125], and Bicici and Yuret [9]. Examples of lexicon-based approaches are the approaches of Veale [130], and the application of the lexicon-based measure of Hirst and St. Onge [39] in SAT, that can be found in the work of Turney [123]. Hybrid approaches have been applied in SAT through the application of the measures of Resnik [91] and Lin[59] that can also be found in the work of Turney [123]. In order for the reader to understand the difficulty of answering SAT questions, we must point out that the average US college applicant scores 57% [126], while the top corpus-based approach scores 56.1% [123], the top lexicon-based scores 42% [130], and the top hybrid scores 33.2% [91].

Another way of categorizing the approaches that measure semantic similarity in analogy tasks is to distinguish among attributional and relational similarity measures [36].[1] Representative approaches of the first category are lexicon-based approaches, while examples of relational similarity measures can be found in approaches based on Latent Relational Analysis (LRA) [123]. It is of great interest to point out that LRA-based approaches, like the LRME algorithm proposed recently by Turney [124], are superior in finding word analogies to attributional similarity approaches. This fact, supported by the experimental findings in [123]. Relational similarity approaches may perform better in the SAT analogy task, but still, as shown later in the experiments we conducted in other applications, like paraphrasing, lexicon-based measures can outperform LRA-based approaches in other tasks.

Semantic relatedness (SR) between words, as applied in Omiotis, can be exploited to solve the word analogy task. The aim of word analogy is, for a given pair of words w_1 and w_2, to identify the series of semantic relations that lead from w_1 to w_2 (semantic path). In the SAT test, the target pair (w_1,w_2) and candidate word pairs (w_{1k},w_{2k}), with k usually being from 1 to 5, are processed in order to find each pair's semantic path. The aim is to locate the pair k, whose elements have the most similar semantic path to that of (w_1,w_2). We are trying to solve this problem by employing

[1]Two things, X and Y, are attributionally similar when the attributes of X are similar to the attributes of Y. Two pairs, A:B and C:D, are relationally similar when the relations between A and B are similar to the relations between C and D.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

two criteria. First, we compare the k pairs to the target pair, and pick the candidate pair whose elements seem to be connected by a semantic relation that is most similar to that of the target pair. However, when the most similar relation is not obvious, we examine all the 6 pairs together in order to find the slight differences between the words comprising each pair. We attempt to model this using SR in a twofold manner: we use SR to measure both the horizontal and the vertical analogy between the target pair and the possible candidate pairs. To capture what we call horizontal analogy between the target pair of words and a candidate pair, we measure the difference of the SR score, of the members of each pair as follows:

$$s_1(w_{1k}, w_{2k}) = |SR(w_1, w_2) - SR(w_{1k}, w_{2k})| \tag{5.1}$$

Essentially, s_1 expresses the horizontal analogy of the candidate pair (w_{1k}, w_{2k}) with the given pair (w_1, w_2), meaning their similarities in the paths connecting the words of the two pairs. Similarly, we capture the notion of the vertical analogy between the two pairs, meaning the similarity of the paths connecting the first word of the target pair, with the first word of a candidate pair (and the same for the second word) by computing the difference of the SR scores among the two word pairs, as follows:

$$s_2(w_{1k}, w_{2k}) = |SR(w_1, w_{1k}) - SR(w_2, w_{2k})| \tag{5.2}$$

Finally, we rank candidates depending on the combined vertical and horizontal analogy the have with the given pair, according to the following equation:

$$s(w_{1k}, w_{2k}) = \frac{s_1(w_{1k}, w_{2k}) + s_2(w_{1k}, w_{2k})}{2} \tag{5.3}$$

Eventually, we select the candidate pair with the maximum combined score, taking into account both aspects (horizontal and vertical) of analogy between the given and the candidate pairs.

The intuition behind the selection of the these two scores for handling the SAT test, is the following. The order of the words in the pairs (both target and candidates) is not random. Usually, given a pair (w_1, w_2), and a the candidate pairs (w_{1k}, w_{2k})

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

the test is solved if one can successfully find the analogy: w_{1k} is to w_{2k} what w_1 is to w_2. From this perspective, s_1 and s_2 try to find the candidate pair that best aligns with the target pair. Figure 5.1 illustrates these two types of analogies (horizontal and vertical) for an example SAT question.

Paraphrasing

The performance of document processing applications relying on natural language processing may suffer from the fact that the processed documents might contain lexically different, yet semantically related, text segments. The task of recognizing pairs of text segments, with identical or almost identical semantics, which is better known as paraphrase detection, is challenging and difficult to solve, as shown in the work of Mihalcea et al. [66], and Pasca [85]. The task itself is important for many text related applications, like summarization [38], information extraction [104] and question answering [84]. We demonstrate the usefulness of the Omiotis measure in paraphrasing detection using the Microsoft Research Paraphrase Corpus [29]. The application of Omiotis to paraphrase detection is straightforward: given a pair of text segments, we compute the Omiotis score between them, using equation 3. Higher values of Omiotis for a given pair denote a stronger semantic relatedness between the two text segments. The task is now reduced to defining a threshold for Omiotis values, above which pairs will be classified as paraphrases. Since no type of training is used in the computation of Omiotis, and since Omiotis values are in $[0, 1]$, we have selected the 0.5 as a threshold value.

Document Similarity

In order to assess how well Omiotis approximates human judgments in document-to-document similarity, we have conducted experiments on a corpus of 50 news documents, taken from the Australian Broadcasting Corporation's news mail service. One main difference from the paraphrase task is that the answer concerning text relatedness is not binary. In this test [57], for each possible pair from the collection of

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

the 50 documents, 10 different human judgements were given, that rated the similarity between documents from 0 to 5, with 5 corresponding to maximum similarity. Inter-rater agreement correlation is about 0.6. Furthermore, an assessment of the 50 documents conducted by Lee et al. [57] against a standard corpus of five English texts, using four models of language (log-normal, generalized inverse Gauss-Poisson, Yule-Simon and Zipfian), showed that the document set is within the normal range of English text for word frequency spectrum and vocabulary growth. Thus, the used collection can be regarded as representative of normal English texts. The use of Omiotis in document similarity is again a straightforward application of equation 3, for all document pairs.

Text Classification

As an additional task to evaluate the ability of Omiotis to measure text relatedness, we embed its core Semantic Relatedness (SR) measure in the text classification task. Several means of embedding semantic information in the text classification task have been considered in the past. A standard methodology is to construct a semantic kernel and embed it in a support vector machines classifier [106, 63, 7, 10, 12]. In the following we present a new GVSM based on a semantic smoothing kernel, that incorporates noun information from WordNet (WSD information for nouns based on the compactness disambiguation method, and the senses hypernyms). Furthermore, we present the incorporation of SR in an existing semantic smoothing kernel. Both models for text classification are evaluated in section 5.2.2 using the Reuters classification data set.

In previous work [63], we presented a Generalized Vector Space Model (GVSM), based on the semantic smoothing kernel introduced in [106]. In that work we introduced a means of embedding WSD information into the classification task (using SVM). Since we aim at embedding WSD in the SVM classifier, we require the definition of a kernel that captures the semantic relations provided by the used hierarchical thesaurus (HT). To the best of our knowledge, the only previous other approaches that define a semantic kernel based on a HT are [106] and [10]. The formal definition of the kernel in [106] is given in definition 7.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Definition 7 *A Semantic Smoothing Kernel between two documents d_1, d_2 is defined as $K(d_1, d_2) = d_1 P' P d_2 = d_1 P^2 d_2$, where P is a matrix whose entries $P_{ij} = P_{ji}$, represent the semantic proximity between concepts i and j.*

The elements of the similarity matrix P are obtained by using a HT similarity measure. The Semantic Smoothing Kernels have similar semantics to the GVSM model defined in [134]. A kernel definition based on the GVSM model is given in definition 8.

Definition 8 *The GVSM kernel between two documents d_1 and d_2 is defined as $K(d_1, d_2) = d_1 D D' d_2$, where D is the term document matrix.*

The rows of matrix D in the GVSM kernel contain the vector representation of terms, used to measure their pairwise semantic relatedness. The Semantic Smoothing Kernel has similar semantics. The Semantic Smoothing Kernel between two documents $K(d1, d2) = d_1 P^2 d_2$, can be regarded as a GVSM kernel, where the matrix D is derived by the decomposition $P^2 = DD'$ (the decomposition is always possible, since P^2 is guaranteed to be positive definite). The rows of D can be considered as the vector representation of concepts, used to measure their semantic proximity. Semantic Smoothing Kernels use P^2 and not P, because P is not guaranteed to be positive definite.

The kernel we define is based on the general concept of GVSM kernel and uses the semantics of the HT. The use of hypernyms for the vector space representation of the concepts of a HT, enables the measurement of semantic distances in the vector space [62]. More precisely, given a tree HT, there exists a weight configuration for the hypernyms, such that standard vector space distance and similarity measures are equivalent to popular HT distances and similarities. This is explained in the following propositions.

Proposition 1 *Let O be a Tree HT. If we represent the concepts of O as vectors containing all their hypernyms, then there exists a configuration for the weights of the hypernyms such that the Manhattan distance (Minkowski distance with $p = 1$) of any two concepts in vector space is equal to the Jiang-Conrath measure [45] in the HT.*

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Proposition 2 *Let O be a Tree HT. If we represent the concepts of the HT as vectors containing all their hypernyms, then there exists a configuration for the weights of the hypernyms such that the Resnik similarity measure [91] in the HT is equal to the inner product in the vector space.*

The WordNet hierarchical thesaurus is composed by 9 hierarchies that contain concepts that inherit from more than one concept, and thus are not trees. However, since only 2.28% of the concepts inherit from more than one concept [28], we can consider WordNet's as being close to trees. From the above we conclude that, if we construct a matrix D where each row contains the vector representation of each sense containing all its hypernyms, the matrix DD' will reflect the semantic similarities that are contained in the HT. Based on D, we move on to define the kernel between two documents d_1, d_2, based on the general concept of GVSM kernels as $K(d_1, d_2) = d_1 DD\prime d_2$. In our experiments we have used various configurations for the rows of D. More precisely, we have considered the vector representation of each concept to be extended with a number of hypernyms (noun POS considered only). We have experimented with several such varying numbers of hypernyms. The argument for using only a limited number and not all hypernyms is that the similarity between hypernyms close to the root of the HT is considered to be very close to 0. The potential of the use of hyponyms was explored as well. The kernel that we finally utilize in our experiments is a combination of the inner product kernel for terms with the concept kernel $K(d_1, d_2) = K_{terms}(d_1, d_2) + K_{concepts}(d_1, d_2)$. This GSVM kernel was embedded into the current version of SVMLight [46] and replaced the standard linear kernel used for document classification with sparse training vectors. The kernel defined implies a mapping from the original term and concept space, to a space that includes the terms, the concepts and their hypernyms. The kernel can be considered as the inner product in this feature space.

Besides the created GVSM kernel for text classification, we have also created an additional semantic smoothing kernel based on the work in [10]. The need for a GVSM kernel, defined in the previous paragraphs, instead of the more straightforward semantic smoothing kernel, originated from the fact that we could not initially (during earlier stages of this work) obtain a matrix P, whose entries $P_{ij} = P_{ji}$ would denote

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

the semantic proximity between concepts i and j. This caveat was due to the fact that we did not use any measure of semantic relatedness between terms at that time, so that we could obtain $P_{i,j}$ for every (i, j) pair. Instead, we used the compactness-based WSD algorithm (section 3.2) and embedded the WordNet concepts along with a defined number of their hypernyms in a GVSM representation of the documents.

Having defined Omiotis however, we were then able use a semantic smoothing kernel that embeds the SR score (see definition 6), as a means for computing P_{ij}. In parallel, wanted to keep the lexical information of the terms in the corpus (i.e., their TF-IDF weights). Thus, we used the kernel proposed by Basili et Al. [7], further discussed and evaluated in [10, 12]. The used kernel is explained in the following: Let d_1 and d_2 be two documents of the document collection. Their similarity is defined as shown in the next equation.

$$K(d_1, d_2) = \sum_{w_1 \in d_1, w_2 \in d_2} (\lambda_1 \lambda_2) \times SR(w_1, w_2) \tag{5.4}$$

where λ_1 and λ_2 are the weights (TF-IDF values in our case) of the words w_1 and w_2 in d_1 and d_2 respectively, and SR is our measure of semantic relatedness between a pair of terms.[2] Then, the kernel can be embedded directly in the SVM classifier. Regarding the SVM implementation, we use SVMLight [46], and for the implementation of the kernel we use a semantic kernel extension for SVMLight.[3]

Text Retrieval

Synonymy (many words per sense) and polysemy (many senses per word) are two fundamental problems in text retrieval. Synonymy is related with recall, while polysemy with precision. One standard method to tackle synonymy is the expansion of the query terms with their synonyms. This increases recall, but it can reduce precision dramatically. Both polysemy and synonymy can be captured on the GVSM model in the computation of the inner product between $\vec{t_i}$ and $\vec{t_j}$ (explained in equation 2.3).

[2] In [7] Basili et al. prove that equation 5.4 is a kernel, using the Lin semantic similarity measure [59].

[3] Thanks to Stephan Bloehdorn, the semantic kernel extension is publicly available at `http://www.aifb.uni-karlsruhe.de/WBS/sbl/software/semkernel/`

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

The methodology of constructing such a GVSM for text retrieval is explained in the following.

In the expansion of the VSM model we need to weigh the inner product between any two term vectors with their semantic relatedness. For this purpose we use the semantic relatedness between a pair of terms (SR) defined in definition 6. In equation 2.3, which captures the document-query similarity in the GVSM model, the similarity between terms t_i and t_j is expressed by the inner product of the respective term vectors $\vec{t_i}\vec{t_j}$. Note that $\vec{t_i}$ and $\vec{t_j}$ are in reality unknown. We estimate their inner product by equation 5.5, where s_i and s_j are the senses of terms t_i and t_j respectively, maximizing $SCM \cdot SPE$.

$$\vec{t_i}\vec{t_j} = SR((t_i, t_j), (s_i, s_j), O) \tag{5.5}$$

Since in our model we assume that each term can be semantically related with any other term, and $SR((t_i, t_j), O) = SR((t_j, t_i), O)$, the new space is of $\frac{n \cdot (n-1)}{2}$ dimensions. In this space, each dimension stands for a distinct pair of terms. Given a document vector $\vec{d_k}$ in the VSM TF-IDF space, we define the value in the (i, j) dimension of the new document vector space as shown in equation 5.6.

$$d_k(t_i, t_j) = (TF - IDF(t_i, d_k) + TF - IDF(t_j, d_k)) \cdot \vec{t_i}\vec{t_j}. \tag{5.6}$$

We add the TF-IDF values because any product-based value results to zero, unless both terms are present in the document. The dimensions $q(t_i, t_j)$ of the query, are computed similarly. A GVSM model aims at being able to retrieve documents that not necessarily contain exact matches of the query terms, and this is its great advantage. This new space leads to a new GVSM model, which is a natural extension of the standard VSM. The cosine similarity between a document d_k and a query q now becomes as shown in equation 5.7.

$$cos(\vec{d_k}, \vec{q}) = \frac{\sum_{i=1}^{n} \sum_{j=i}^{n} d_k(t_i, t_j) \cdot q(t_i, t_j)}{\sqrt{\sum_{i=1}^{n} \sum_{j=i}^{n} d_k(t_i, t_j)^2} \cdot \sqrt{\sum_{i=1}^{n} \sum_{j=i}^{n} q(t_i, t_j)^2}} \tag{5.7}$$

where n is the dimension of the VSM TF-IDF space.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Other Applications of Omiotis

Besides the aforementioned applications of SR and Omiotis, for which we provide experimental evaluation in chapter 5, there are also other interesting applications, for which initial experimental evaluation is encouraging. One such application is the use of Omiotis for organizing bibliographical data from paper repositories, like DBLP[4]. In this direction we have applied Omiotis as a paper-to-paper similarity measure based on the paper titles lying in DBLP, from selected conferences (e.g. ECML, ECDL, FOCS, VLDB, etc.). The Omiotis metric can then be applied in text clustering and classification tasks by substituting the traditional document similarity measures (e.g. cosine similarity) with the semantic relatedness measure. Below, we demonstrate Omiotis' incorporation into the k-nearest neighbor (k-NN) classification scheme [27] and into the clustering algorithms of the CLUTO [138] suite. Of course, our measure can be employed by any other data organization method that operates upon the notion of word or instance similarities. To demonstrate how Omiotis can be explored by the above schemes, we examine the case of publication titles' classification and clustering, respectively.

In the original k-NN classification algorithm, a new instance is classified by a majority vote of its neighbors. Specifically, assume we want to classify instance j to the most suitable class, out of m classes. Assume also that we have available a training set for which we already know the correct classes. Instance j is classified to class c_i -the most frequent class amongst its k nearest neighbors- using the following formula:

$$c_i = \arg\max_{c=1..m} |O_{kc}| \tag{5.8}$$

where $|O_{kc}|$ is the number of training instances that belong to class c_i, with $i = 1, .., m$. The parameter k signifies that we only explore the k nearest neighbors of j every time a classification is made. Based on the above steps, the k-NN classifier assigns texts to their corresponding classes. To classify documents that are represented in the VSM, the cosine similarity or Jaccard's coefficient metrics are employed in order to identify the neighboring training instances. We replace the notion of similarity with that of

[4] http://www.informatik.uni-trier.de/~ley/db/

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

relatedness and we employ the Omiotis measure for deriving the relatedness between semantic aspects of texts.

Regarding Omiotis's incorporation into CLUTO, the suite offers three different clustering algorithms that can be applied either (i) to text instances, or (ii) to any type of instances as long as a similarity matrix between instances exists. The second case is the most straightforward for incorporating Omiotis, since we can use a pre-computed similarity matrix as input to the algorithm instead of modifying its internal mechanism. In particular, we use the *scluster* program of CLUTO, which takes as input the adjacency matrix of instances and the desired number of classes. We also use the adjacency matrices produced by the cosine similarity and the Omiotis measures respectively, and we compare the obtained results. Initial experimental results show a constant improvement of classification and clustering of DBLP paper titles using Omiotis, against the use of the traditional GVSM.

In another application setting, we have embedded the SR measure into a novel keyword extraction algorithm, SemanticRank, that we define below. Identifying the most important terms in a text is of paramount importance for a variety of tasks. Currently, the most widely used keyword extraction method is the TF-IDF weighting scheme that estimates the importance of terms based on their statistical properties (i.e., frequency counts) in the texts in which they appear. Recently, there has been a significant body of research, which demonstrates that supervised methods are more successful than traditional term extraction techniques. For an overview, we refer the reader to the work of [69]. Here, we introduce a new approach for keyword extraction. Our method builds upon the SR measure, and employs a novel algorithm for quantifying the importance of every term in a text.

Our algorithm, named SemanticRank, relies on the semantic relatedness graph of document terms and is mutatis mutandis the TF-IDF scheme in the world of semantics. More specifically, assume a document collection C and a document $d \in$ C with n distinct terms t_i, $i = 1..n$. By relying on the SR measure of definition 6, we can compute the semantic relatedness between every distinct pair of terms ($\frac{n \cdot (n-1)}{2}$ in total) in every document d. Thereafter, we consider a semantic graph where every term constitutes a vertex V_i and every edge E_{ij} the semantic relatedness


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

between terms t_i and t_j. Based on the above, we compute the SemanticRank score for each term by using an adaptation of the well-known PageRank formula [69]. More specifically, the SemanticRank of a term i is determined as:

$$SemanticRank(t_i) = (1-d) + d \cdot \sum_{t_j \in IN(t_i)} \frac{SR(t_i, t_j)}{\sum_{t_k \in OUT(t_j)} SR(t_j, t_k)} SemanticRank(t_j) \tag{5.9}$$

The intrinsic aim of SemanticRank is that it rewards the centrality of terms in the semantic relatedness graph in an analogous manner that PageRank rewards the centrality of nodes in the web graph. Thus, as PageRank promotes the most strongly connected graph nodes, SemanticRank promotes the terms that are the most semantically related to the majority of other terms in the collection graph. Based on the SemanticRank values computed for every term in a text, we can extract the top k terms as the most important keywords for communicating the text semantics. Initial experimentation in an effort to label the clusters produced by bibliographical data from DBLP, again provides very interesting and promising results.

In another direction, in the past we had examined the introduction of WSD information into Web personalization [30]. Web personalization is the process of customizing a web site to the needs of each specific user or set of users. Personalization of a web site may be performed by the provision of recommendations to the users, highlighting/ adding links, creation of index pages, etc. The web personalization systems are mainly based on the exploitation of the navigational patterns of the web sites visitors. When a personalization system relies solely on usage-based results, however, valuable information conceptually related to what is finally recommended may be missed. The exploitation of the web pages semantics can considerably improve the results of web usage mining and personalization, since it provides a more abstract yet uniform and both machine and human understandable way of processing and analyzing the usage data. The underlying idea is to integrate usage data with content semantics, expressed in ontology terms, in order to produce semantically enhanced navigational patterns that can subsequently be used for producing valuable recommendations. The work in [30] was a proposal of a semantic web personalization system, focusing on WSD techniques which can be applied in order to semantically

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

annotate the web sites content. Under the same scope, we aim at applying Omiotis to automatically annotate Web pages with semantic information from WordNet. New WordNet senses can also aid to construct and evolve a domain ontology of the considered Web pages [118] helping to crate thematic user profiles.

5.2 Experimental Evaluation

The experimental evaluation of Omiotis is two-fold. First, we test the performance of the semantic relatedness measure for a pair of words in four benchmark data sets, namely the Rubenstein and Goodenough 65 word pairs [96] (R&G), the Miller and Charles 30 word pairs [72] (M&C), the Word-Similarity-353 collection [32] (353-C) comprising 353 word pairs, and the SAT Analogy questions, comprising 374 test questions. Second, we evaluate the performance of Omiotis in four text related tasks, namely the paraphrase detection task, using the Microsoft Research Paraphrase Corpus [29], a document similarity task [57], the document classification task using the Reuters-21578 document collection (ModApte split) and the Amazon data set, and finally the text retrieval task, using three TREC collections.

5.2.1 Word-to-Word Semantic Relatedness

Comparison of the Semantic Relatedness Measure to Human Perception

For the evaluation of the proposed semantic relatedness measure between two terms we used three widely used data sets in which human subjects have provided scores of relatedness for each pair. A kind of "gold standard" ranking of related word pairs (i.e., from the most related words to the most irrelevant) has thus been created, against which computer programs can test their ability on measuring semantic relatedness between words. We compared our measure against ten known measures of semantic relatedness: Hirst and St-Onge (HS)[39], Jiang and Conrath (JC)[45], Leackock and Chodorow (LC)[55], Lin (L)[59], Resnik (R)[91, 92], Jarmasz and Szpakowicz (JS)[44], Gabrilovich and Markovitch (GM)[34], Finkelstein et al. (F)[32], Hughes and Ramage (HR)[41], and Strube and Ponzetto (SP) [113, 88]. In table 5.1 we show the results for

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

	HS	JC	LC	L	R	JS	GM	F	HR	SP	SR
R&G	0.745	0.709	0.785	0.77	0.748	0.842	0.816	*N/A*	0.817	0.56	0.861
M&C	0.653	0.805	0.748	0.767	0.737	0.832	0.723	*N/A*	0.904	0.49	0.855
353-C	*N/A*	*N/A*	0.34	*N/A*	0.35	0.55	0.75	0.56	0.48	0.552	0.61

Table 5.1: Correlations of semantic relatedness measures with human judgements.

all three data sets and for all ten measures, when these were available. The reported numbers are the Spearman correlation of the results of each measure with the gold standard (human judgements). The detailed scores for humans for R&G and M&C data set can be found in [18], while the detailed scores for the (353-C) data set are made available with the collection[5]. The correlations for the R&G and M&C data sets show that SR performs in general better than any other measure of semantic relatedness of any category (knowledge-based, corpus-based or hybrid). To visualize the performance of our measure in a more comprehensible manner, we also present in figure 5.2.1 the relatedness values given by humans for all pairs in the *R&G* and *M&C* data sets, in increasing order of value (left side) and the respective values for these pairs produced using SR (right side). A closer look on figure 5.2.1 reveals that the values produced by SR (right figure) follow a pattern similar to that of the human ratings (left figure). Note that the x-axis in both charts begins from the least related pair of terms, according to humans, and goes up to the most related pair of terms. The y-axis in the left chart is the respective humans' rating for each pair of terms. The right figure shows SR for each pair. The reader can consult [18] to confirm that all the other measures of semantic relatedness we compare to, for the same two data sets, do not follow the same pattern as the human ratings, as closely as our measure of relatedness does (low y values for small x values and high y values for high x). With regards to the SR performance in the 353-C data set, we note that it ranks second, right after the Wikipedia-based measure of Gabrilovich and Markovitch (GM), but surpasses the rest, including the Wikipedia-based method of Strube and Ponzetto (SP). Note also that Omiotis wins both GM and SP in the R&B and M&C data

[5] `http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/`

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

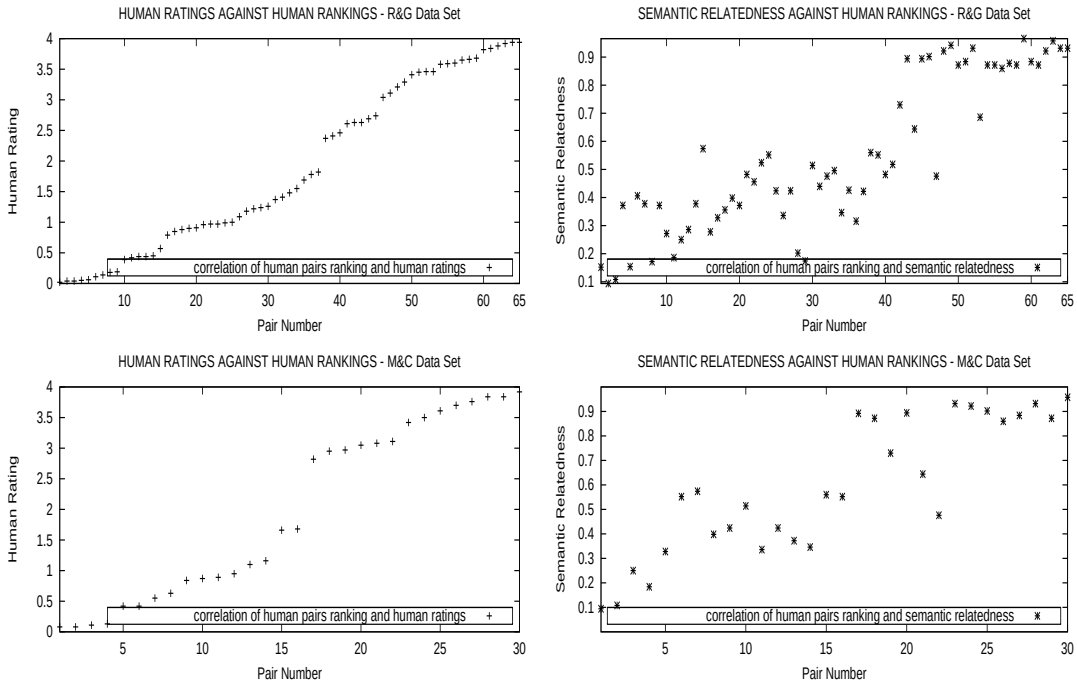

Figure 5.2: Correlation between human ratings and SR in the R&G and M&C data sets.

sets, and that the 353-C data set contains the term pairs of the M&C data set. As a further remark regarding the 353-C collection, we need to add the fact that there are cases where the inter-judge correlations may fall below 65%, while R&B and M&C data sets have inter-judge correlations between 0.88 and 0.95. Finally, regarding the statistical significance of the results, the correlations of SR rankings to the human rankings (0.861,0.855 and 0.61 respectively for the three data sets) denote a significant positive correlation at the 0.99 confidence level.

SAT Analogy Questions

The approach that we chose to evaluate SR in the analogy task is to use the typical benchmark test set employed in the related bibliography, namely the Scholastic Aptitude Test (SAT).[6] It comprises 374 words pairs and for each target pair 5 supplementary pairs of words. The average US college applicant answered correctly only

[6]Many thanks to Peter Turney, for providing us with a standard set for experimentation.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

	RG	JC	L	LC	HS	R	B	V	T	S1	S2	S	UB	NB
Prec.	0.2	0.273	0.273	0.313	0.321	0.332	0.4	0.42	0.561	0.283	0.304	0.34	0.524	0.381

Table 5.2: Precision in the 374 SAT Questions.

57 percent of the questions, which consequently became the upper bound for every machine based approach. In table 5.2, we present the precision on the 374 SAT questions, of nine methods, namely random guessing (RG), Jiang and Conrath (JC)[45], Lin (L) [59], Leacock and Chodrow (LC)[55], Hirst and St.-Onge (HS)[39], Resnik (R)[91], Bollegala et al. (B)[14], Veale (V)[130], and Turney (T) [123]. Furthermore, we present the individual results of S1 (equation 5.1), S2 (equation 5.2) and S (equation 5.3). Towards the direction of combining the answers of S1 and S2 in a different manner than the naive average, we also report the results of an "oracle" that would always choose the correct score among S1 and S2. This is reported in the table as our upper-bound (UB). In an effort to design a learning mechanism that would learn when to select S1 or S2 answers for each SAT question, with the goal to reach our upper-bound, we designed and implemented a simple representation of the SAT questions as training instances. For each SAT question, we created a training instance that has 6 features: the minimum S1 value found for this question (among the five computed values for all the possible pairs), the maximum S1 value, and their difference. We also added the same features regarding S2. We then trained and tested a Naive Bayes classifier [47] using ten-fold cross validation in the 374 SAT questions. The classification is binary (trust S1 or not trust S1, meaning to trust S2), and we used the respective Weka implementation. The results of this experiment are shown in the table as (NB). Finally, we also present the top results ever reported in the literature for the specific data set, which is the LRA method by Turney [123]. This is reported in the table as (T). The results presented in table 5.2 show that S ranks second among all lexicon-based measures loosing only by the measure of Veale (V)[130], which is especially tuned for the SAT test. The method of Bollegala et Al. (B) achieves higher score than SR, but needs training on SAT questions. At this point we have to note that the LRA method (T) needs almost 8 days to process the

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

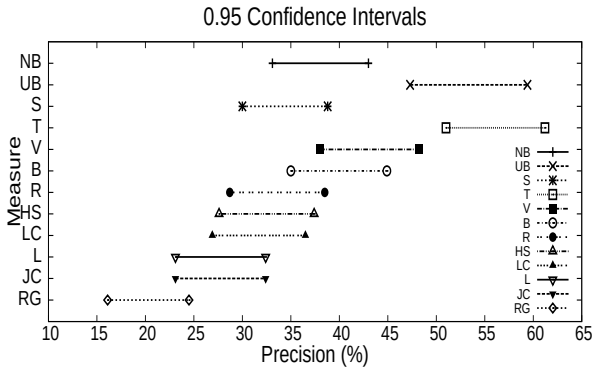

Figure 5.3: 0.95 confidence intervals in the 374 SAT questions.

374 SAT questions [123], (B) needs around 6 hours [14], while SR needs a little less than 3 minutes. The precision of SR in the 374 SAT questions, along with the fact that it needs very little time to execute, given the implementation discussed in the previous chapter, shows that SR can be successfully applied to the SAT task, and it also needs orders of magnitude less execution time, compared to some of the state of the art approaches.

Furthermore, the fact that combining S1 and S2 can reach 52.4% shows that SR can produce very promising results, if a classifier learns successfully how to combine them. The NB results, which are a simple attempt to construct such a learner with few features, shows an important boost in performance of 4.1%. Better feature engineering and more training SAT questions can potentially yield more promising results, as the gap between 38.1% and the upper bound of 52.4% is still large. In all, these results show that our lexicon-based relatedness measure is very efficient even when used in a task which it has not been designed for. In addition, as results show, it can be a useful tool for designing a sophisticated solution especially for the SAT tests.

Finally, regarding the statistical significance of the results, figure 5.3 shows the 0.95 confidence intervals for all compared measures. As shown, S matches the performance of all lexicon-based measures, while the upper bound of combining S1 and S2 can lead to a performance that matches the top reported results in the specific 374 SAT questions by Turney [123].

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

	Corpus-based			Knowledge-based							
	PMI-IR	LSA	STS	JC	LC	Lesk	L	WP	R	Com.U	Omiotis
ERR	13	8.67	11.93	11.27	11.84	11.27	11.27	10.4	10.4	14.16	14.69

Table 5.3: Error Reduction Rates (%) from the standard vectorial model in the paraphrase task.

5.2.2 Text-to-Text Semantic Relatedness

Paraphrase Task

In order to evaluate how well Omiotis measures the semantic relatedness between texts, we decided to run the paraphrase recognition task on the test pairs of the Microsoft Research Paraphrase Corpus [29]. From the original data set, containing both training and test pairs, we run experiments only on the 1725 test pairs of text segments, which have been collected from news sources on the Web over a period of 18 months. For each pair, human subjects have determined whether any of the two texts in the pair consists a paraphrase of the other (the direction is not an issue). The inter-judge agreement between annotators has been 83%.

For this task we computed Omiotis between the texts of every pair and marked as paraphrases only those pairs with Omiotis greater than a threshold. The threshold was set to 0.5 since the measure's values range from 0 to 1. We compare the performance of Omiotis against all the other measures of semantic relatedness measuring the Error Reduction Rate (ERR) compared to the baseline Vector Space Model using cosine as the similarity measure. Table 5.3 shows the reported ERR for Omiotis, as well as for JC[45], LC[55], L[59], R[91, 92] and LSA[32]. We have also added ERR for the simple Lesk measure (Lesk) [58], the Wu and Palmer measure (WP) [135], the PMI-IR corpus-based measure suggested by Turney [122], the STS corpus-based measure proposed by Islam and Inkpen (using the reported results for the same threshold of 0.5) [43] and two combined measure proposed by Mihalcea et al. [66] (Com.U). The results indicate that Omiotis surpasses all the compared knowledge-based and corpus-based measures in the paraphrase task, providing an error reduction rate of 14.69% to the vector space model with cosine similarity.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

	COS	CFM	RM	DFM	LSA	GM	Omiotis
Correlation	0.27	$0.22 - 0.49$	$0.32 - 0.49$	$0.03 - 0.14$	0.6	0.72	0.4427

Table 5.4: Correlations to human judgements for the 50 documents data set.

Document Similarity Task

The second text related task for Omiotis, is the document similarity task, discussed in section 5.1. The data set comprises 50 documents and the number of words per document varies from 51 to 126. For all possible pairs of documents, we have human judgments of similarity. The inter-rater correlation, which can be considered as the human performance and an upper bound for the task, is 0.605. We computed Omiotis for all the pairs and measured the correlation between our results and the human judgments. We compared Omiotis against six other measures, namely a baseline text similarity method using the vector space model as documents representation, cosine as documents similarity measure and TF-IDF as terms' weights (COS), the Common Features Model (CFM)[56], which assumes that similarity is measured by the proportion of common features (terms in our case); Tversky's Ratio Model (RM)[127], which measures similarity as the ratio of common to common and distinctive features; the Distinctive Features Model (DFM), which is a special case of Tversky's contrast model [127] and is based more on the dissimilarity of the compared documents; the LSA method of Lee et Al. (LSA)[57]; and finally, the ESA-Wikipedia method of Gabrilovich and Markovich (GM)[34]. Table 5.4 shows the reported results, in terms of correlation to the human judgements. For CFM, RM and DFM, the ranges in correlation have been obtained by using different setups of the models [57]. More specifically, the lower correlations were obtained without considering n-grams, while higher correlations were achieved when using 7-, 8-, or 9-grams. The results reveal the following interesting findings: (a) Omiotis surpasses the COS and DFM models, while it is also better in the majority of the setups of the CFM and RM models. Note also that CFM and RM need tuning to perform their highest correlations. (b) RM, CFM, Omiotis, LSA and GM are significantly better than COS and DFM. (c) LSA and GM achieve top correlations. In all, Omiotis performs well on the 50 documents

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

data set, even though it does not require any type of training; Note also that LSA requires extensive tuning to select the optimal number of dimensions for the singular value decomposition. The reader is also reminded that the LSA-based approaches were outperformed by Omiotis in the paraphrase task.

Document Classification Task

In this section we present the results of our experimental evaluation in the document classification task, for the GVSM kernel based on the compactness WSD measure and the semantic smoothing kernel based on the SR measure. Regarding the evaluation of the GVSM kernel, we have conducted experiments in the Reuters-21578 data set and a data set from amazon.com using Amazon's publicly available Web interface. Reuters-21578 is a compilation of news articles from the Reuters newswire in 1987. We include this collection mostly because it has become a standard benchmark in document classification. We conducted experiments on the two largest categories, namely *acquisitions* and *earnings*, using the split in training and test documents of [11]. This split yields a total of $4,436$ training and $1,779$ test documents for the two categories. We extracted features from the mere article bodies, and hiding from the classifier any direct hints as to the actual topic (e.g., keywords tags). Standard term-based classifiers achieve very high accuracy on Reuters-21578 given a sufficiently large training set. The interesting point in using this collection is to compare known results with the behavior of our approach at various smaller training set sizes. Regarding the Amazon data set, this site promotes books which are classified in categories. From that taxonomy, we selected all the available editorial reviews for books in the three categories Physics, Mathematics, and Biological Sciences, with a total of $6,167$ documents. These reviews typically contain a brief discussion of a books content and its rating. Since there is a high overlap among these topics vocabulary and a higher diversity of terms within each topic than in Reuters, we expect this task to be more challenging.

We POS-annotated both the Reuters and Amazon collections (using the Stanford tagger [116]) and we restricted the disambiguation step to matching noun phrases in WordNet, because compactness-based disambiguation can only handle nouns. Since

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

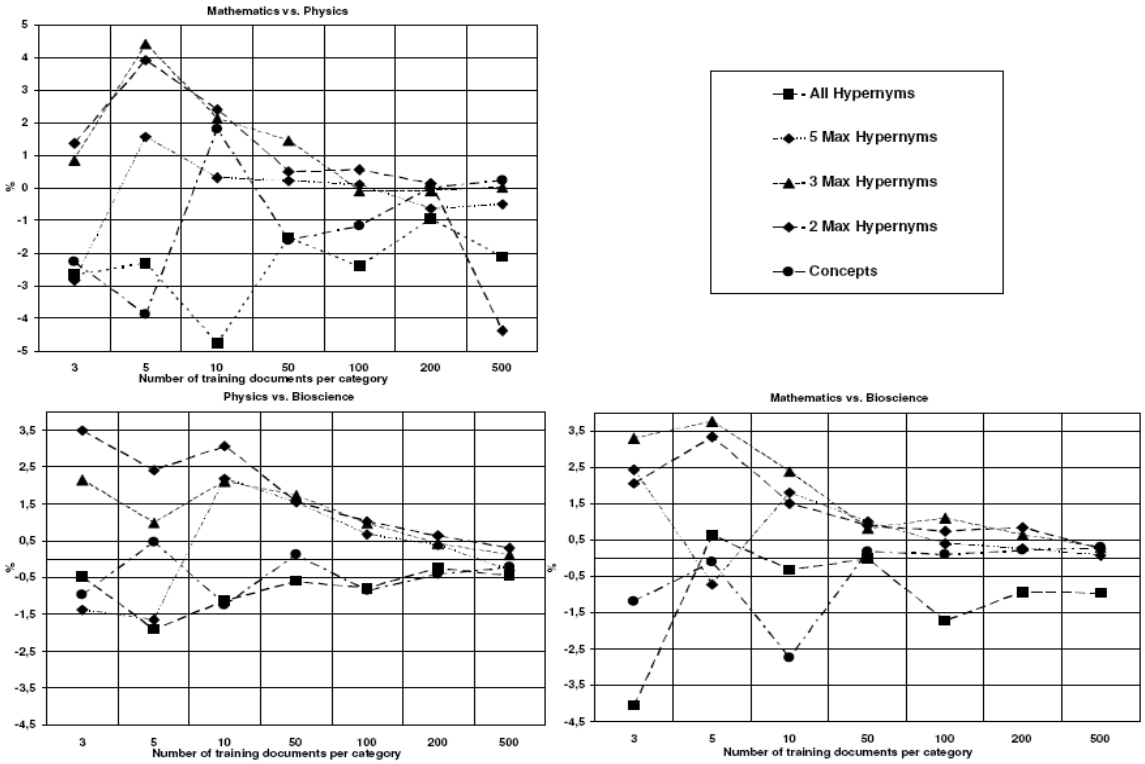

Figure 5.4: Relative Improvement of F-measures scores for various Similarity Configurations in the Amazon Topics.

WordNet also contains the POS information for each of its concepts, POS document tagging significantly reduces the amount of choices for ambiguous terms and simplifies the disambiguation step. For example the term *run* has 52 (!) distinct senses in WordNet out of which 41 are tagged as verbs. We first consider adjacent noun phrase tokens in a small window of up to a size of 5 into dictionary lookups in WordNet before the disambiguation step takes place. If no matching phrase is found in WordNet within the current window, the window is moved one token ahead. This sliding window technique enables us to match any multi-word noun terms known in WordNet, whereupon larger phrases are typically less ambiguous. Non-ambiguous terms can be chosen directly as safe seeds for the compactness-based disambiguation step. Note that we did not perform any feature selection methods such as Mutual Information or Information Gain [61] prior to training the SVM. The binary classification tasks

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

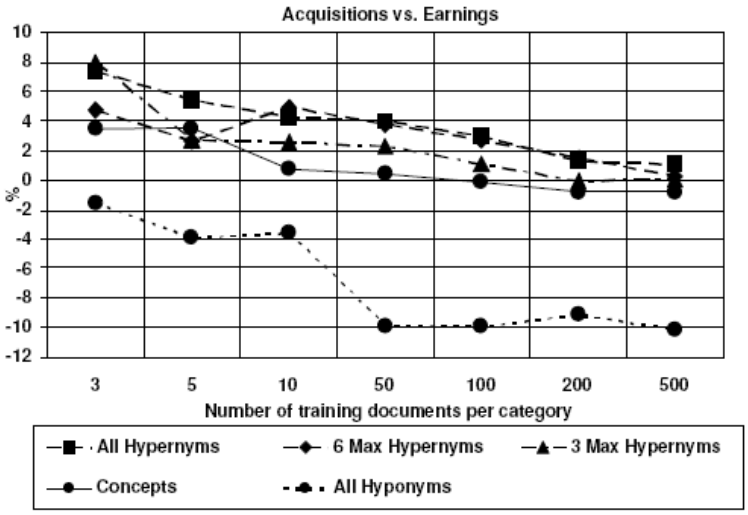

Figure 5.5: Relative Improvement of F-measures scores for various Similarity Configurations in the Reuters Topics.

were performed after forming all pairs between the three Amazon topics, and one pair between the two largest Reuters-21578 topics. The parameters setting for the compactness-based WSD was $W3L0$, since it achieved high percision and performed in a stable manner during the WSD evaluation experiments in the WSD benchmark corpora. Our baseline was the F-Measure [61] arising from the usage of term features only (e.g., not considering WordNet concepts as well). The baseline competed against the embedding of the term senses, whenever disambiguation was possible, and their hypernyms/hyponyms into the term feature vectors, according to the different GVSM kernel configurations shown in figures 5.4 and 5.5. We varied the training set sizes between 3 and 500 documents per topic. For each setup, in figures 5.4 and 5.5 we report the differences of the macro-averaged F-Measure between the baseline and the respective configurations, using 10 iterations for each of the training set sizes to reduce the degree of result variances due to a few document outliers. For more than 500 documents, all our experiments indicate a convergence in results between the concept-based classifier and the text classifier based on term features only. For each run, the training documents were selected randomly following a uniform distribution. Since there is no split into separate documents for training and testing given in the

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Amazon collection, we performed cross-validation runs over the whole set, each using all the remaining documents for the test phase.

The results demonstrate that the use of compactness-based WSD and our kernel function, based on a small number of hypernyms increases consistently the classification quality especially for small training sets. In some cases, as the number of hypernyms increases we observe a performance deterioration which in some cases falls below the term-based classification. The variance in the number of hypernyms needed to achieve better performance can be explained by the fact that we did not employ a hypernym weighting scheme. Thus, when semantically correlated categories are considered, such as Maths/Physics in the Amazon data, then the use of all the hypernyms with equal weights would result in many documents belonging to the Physics category to have a high similarity to documents of Maths category, degrading the performance of the classification algorithm.

As a next step, we compare the GVSM kernel with the semantic smoothing kernel using SR. We selected the best set up to compare with, which was the embedding of at most 6 hypernyms in the GVSM, for every disambiguated noun. Note that the previous GVSM requires a noun disambiguation step prior to execution, increasing the computational complexity of the corpus pre-processing. The conducted experiments for this comparison use only the two largest Reuters categories, namely *acquisitions* and *earnings*. At first, binary classification was performed using a linear kernel with SVMLight, and weighting term features with their TF-IDF values. For this experiment, the computation of the TF-IDF values was restricted only by keeping the documents of the two used categories (TF values are not affected). Again, we did not use any feature selection methods, like Mutual Information or Information Gain.

In order to comply with the previous experimental results, we varied the number of documents used for training between 3 and 500 documents per topic. For each setup, we measured the *macro-averaged F-Measure*, using 10 iterations for each of the training set sizes, to reduce the degree of result variances due to a few document outliers. In figure 5.6 the results from the conducted experiments are shown. The top figure shows the absolute increase in the macro-averaged F1 values from the baseline, for both the GVSM and the semantic kernel and for all training set sizes.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

The bottom figure shows the exact macro-averaged F1 values for the three classifiers and for all training set sizes. The results clearly show that all three classifiers achieve very high macro F1 scores, all above 75%. The proposed semantic kernel shows an improvement for both the baseline and the compared GVSM. Especially, if we focus on the experimental results when small training size sets were used, the semantic kernel boosts macro F1 scores up to 6.25% compared to the baseline, and almost 2% compared to the GSVM. Overall, the semantic kernel uses richer semantic information than the compared GVSM, which is restricted only to nouns and the use of their hypernyms. The SR used in the semantic kernel can produce term to term semantic relatedness values that can aid the document classification task more than the GVSM, especially when the available training size sets are small.

To demonstrate further the classification boost that the used kernel can achieve, we have also conducted experiments on the ten largest Reuters categories, namely *earn, acq, crude, trade, money-fx, interest, money-supply, ship, sugar*, and *coffee*. The experiments were conducted following the procedure in [10] so that the results are comparable. The experimental setup in this case lies in selecting for each execution randomly a small percentage from the Reuters training set (2%, 3%, 4% and 5%) and applying classification to the full test set of the designated Reuters categories. The experiment has been repeated 10 times for each training subset size, and binary classification is applied each time for each category (one-against-all classification strategy). The final F1 scores are averaged to compute the macro-F1 score for each training size. In figure 5.7 we present our results. In the top figure we show the difference (in absolute percentage points) in the macro-F1 scores of our kernel from the baseline, which is the linear kernel. We also show the same difference for the best performed kernel of Bloehdorn et al. [10]. Note that the two kernels are the same, and the difference lies in the underlying similarity measure between terms that produces the term-to-term similarity matrix. In [10] their best setup is with the use of the similarity measure of Lin [59], and the differences depicted in the upper figure represent that specific setup. In the bottom figure we show the precise macro-F1 values for our kernel and the linear kernel. At this point we have to note that we report much lower macro-F1 scores for the linear kernel, than the scores reported

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

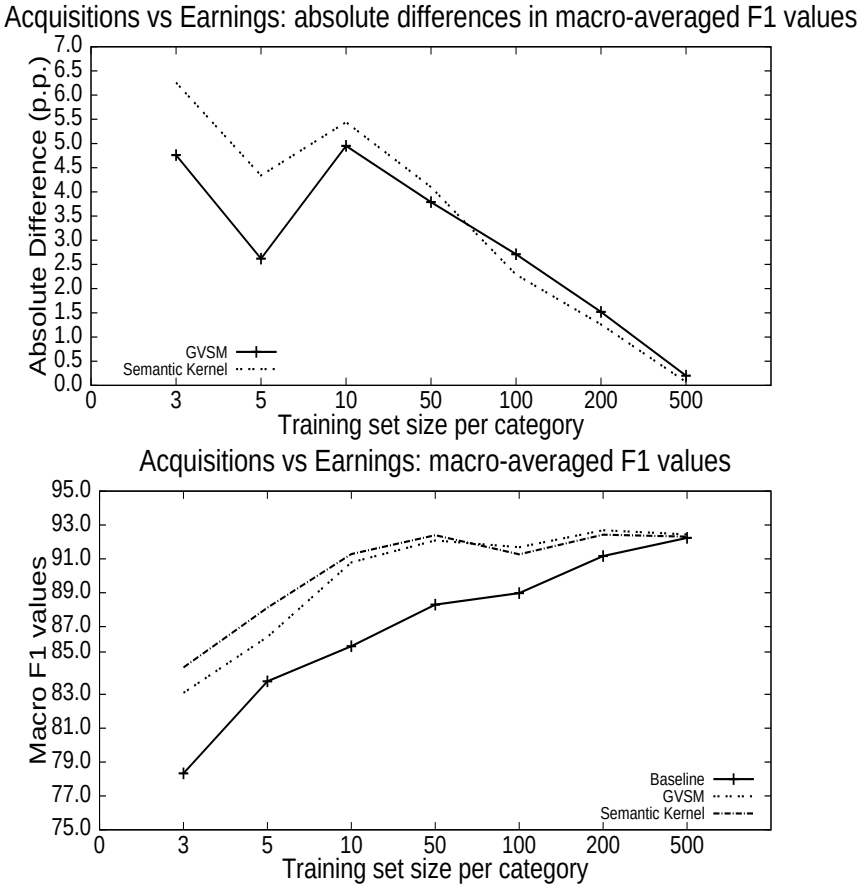

Figure 5.6: Absolute improvements of macro F1 values and exact macro F1 values for the *Acquisitions vs Earnings* experiment.

in [10]. This must be due to the different pre-processing conducted in the ModApte split (stopwords, TF-IDF formula variation, etc.). Furthermore, we have not removed other features than the ones in the used stopword list (i.e., features appearing less than 5 times in the whole dataset). The results show an improvement to the linear kernel macro-F1, of up to 9.11% when using only 2% of the training set. We can also see a small improvement over the Lin Kernel for the very small training subsets. Our findings align to the conclusions extracted in [63] and [10], with regards to the fact that semantic information from WordNet can boost the classification task when there is limited availability in training data. Finally, we have also discovered through the experimental evaluation, as in [10], that when the training set becomes larger,

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

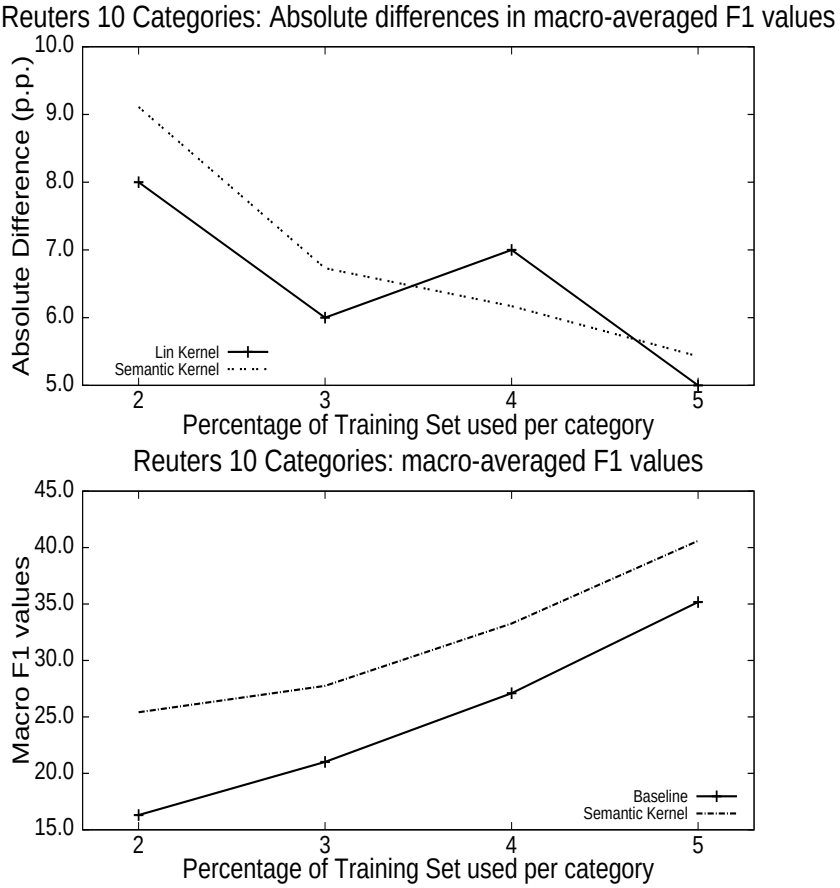

Figure 5.7: Absolute improvements of macro F1 values and exact macro F1 values for the 10 largest Reuters categories experiment.

the performance of the semantic kernel converges to the results of the linear kernel, as also shown in figure 5.6, where larger training sets are used (i.e. 500 training documents per category), and in few cases the semantic kernel even deteriorates the performance by a small percentage of almost 1%.

Regarding the statistical significance of the results, we have performed a macro t-test [136] for the macro-F1 values of the linear kernel and our semantic kernel, for both experiments. In all cases, besides the training size of 500 documents per category in our first experiment, the differences in the macro F1 values are statistically significant at the 0.95 confidence level. Furthermore, we also made the same test for the macro-F1 values of our semantic kernel and the GVSM kernel of the first experiment. We

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

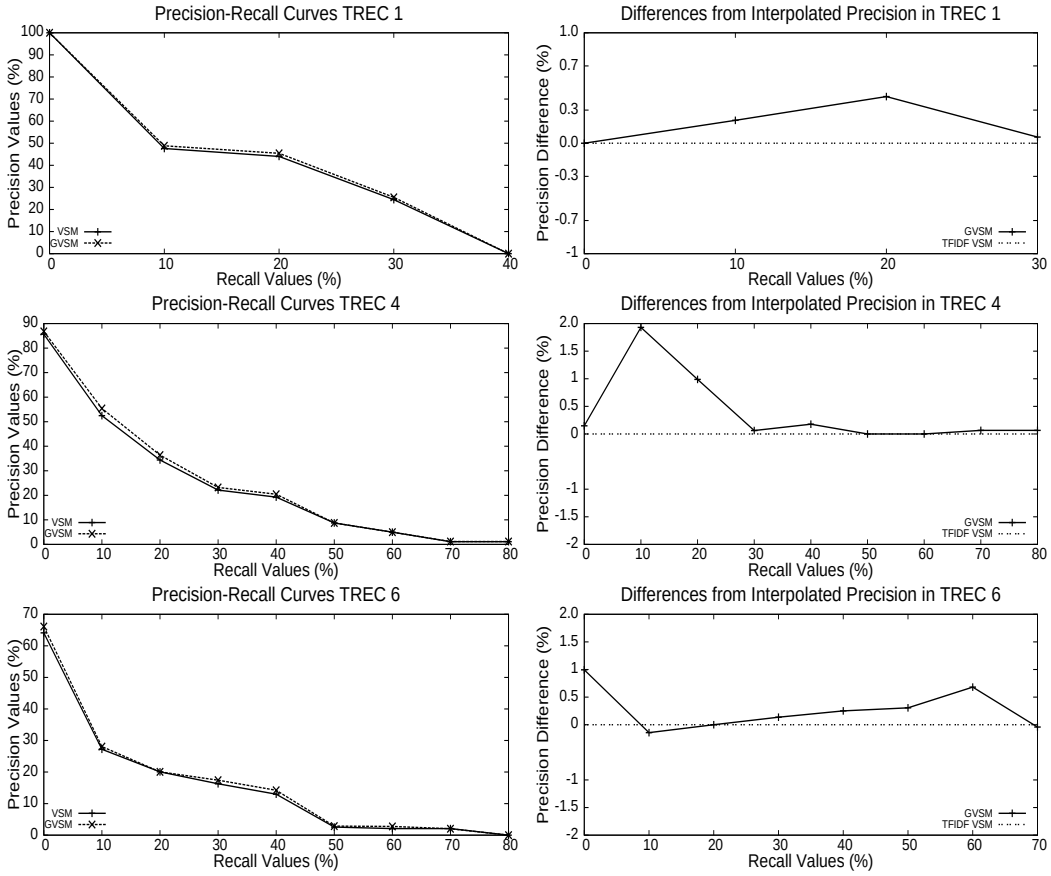

Figure 5.8: Interpolated precision recall curves and differences (percentage points) from the baseline in interpolated precision.

found that the differences in the macro-F1 values are statistically significant at the 0.95 confidence level, only for the training sizes of 3 and 5 documents per category. The same test could not be done for the differences with the Lin kernel, because the F1 values for all the individual categories, necessary for performing the macro t-test, are not publicly available for this kernel.

Text Retrieval

For the evaluation of the proposed GVSM model, we have experimented with three TREC collections [7], namely TREC 1 (TIPSTER disks 1 and 2), TREC 4 (TIPSTER

[7] `http://trec.nist.gov/`

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

disks 2 and 3) and TREC 6 (TIPSTER disks 4 and 5). We selected those TREC collections in order to cover as many different thematic subjects as possible. For example, TREC 1 contains documents from the Wall Street Journal, Associated Press, Federal Register, and abstracts of U.S. department of energy. TREC 6 differs from TREC 1, since it has documents from Financial Times, Los Angeles Times and the Foreign Broadcast Information Service.

For each TREC, we executed the standard baseline TF-IDF VSM model for the first 20 topics of each collection. Limited resources prohibited us from executing experiments in the top 1000 documents. To minimize the execution time, we have indexed all the pairwise semantic relatedness values according to the SR measure, in a database, whose size reached 300GB, and integrated this infrastructure into the Terrier retrieval platform, as explained in section B.3. Thus, the execution of the SR itself is really fast, as all pairwise SR values between WordNet synsets are indexed. For TREC 1, we used topics $51 - 70$, for TREC 4 topics $201 - 220$ and for TREC 6 topics $301 - 320$. From the results of the VSM model, we kept the top-50 retrieved documents. In order to evaluate whether the proposed GVSM can aid the VSM performance, we executed the GVSM in the same retrieved documents. The interpolated precision-recall values in the 11-standard recall points for these executions are shown in figure 5.8 (left graphs), for both VSM and GVSM. In the right graphs of figure 5.8, the differences in interpolated precision for the same recall levels are depicted. For reasons of simplicity, we have excluded the recall values in the right graphs, above which, both systems had zero precision. Thus, for TREC 1 in the y-axis we have depicted the difference in the interpolated precision values (%) of the GVSM from the VSM, for the first 4 recall points. For TRECs 4 and 6 we have done the same for the first 9 and 8 recall points respectively.

As shown in figure 5.8, the proposed GVSM may improve the performance of the TFIDF VSM up to 1.93% in TREC 4, 0.99% in TREC 6 and 0.42% in TREC 1. This small boost in performance proves that the proposed GVSM model is promising. There are many aspects though in the GVSM that we think require further investigation, like for example the fact that we have not conducted WSD so as to map each document and query term occurrence into its correct sense, or the fact that

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

the weighting scheme of the edges used in SR is generated from the distribution of each edge type in WordNet, while there might be other more sophisticated ways to compute edge weights. We believe that if these, but also more aspects discussed in the next section, are tackled, the proposed GVSM may improve more the retrieval performance.

5.3 Discussion of the Experimental Evaluation

In this chapter, we have experimentally evaluated a new measure of text semantic relatedness, Omiotis. The major strength of this measure lies in the formulation of the semantic relatedness between words. Experimental evaluation showed that our measure approximates human understanding of semantic relatedness between words better than previously proposed measures. The combination of path length, nodes' depth and edges' type in a single formula allowed us to apply our semantic relatedness measure to different text-based tasks with promising performance. More specifically, the SR measure surpassed all state of the art measures in word-to-word tasks and the Omiotis measure performed significantly well in the paraphrase and text classification task. Although, the results in the word analogy task are satisfactory, since no special tuning has been performed, we are sure that there is still place for improvement.

From the set of our experiments, we conclude that our measure can be easily applied to several text related tasks. More specifically, we have introduced two different methodologies of embedding semantic information from WordNet in the text classification task. The first method, a GVSM kernel, that is based on the disambiguation information produced by the compactness-based WSD introduced in section 3.2, shows statistically significant improvement in two data sets over the linear kernel of SVM. The second method, a semantic smoothing kernel that uses SR, shows additional improvement over the linear kernel, and in many cases statistically significant improvement over the GVSM kernel. The results of this experiment in all cases revealed that text classification improves significantly if semantic information from WordNet is used, especially when the training set is small.

Finally, we showed that the proposed measure of semantic relatedness can be

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

applied to text retrieval through integration with the Terrier platform. This achievement is much due to the infrastructure and the implementation we have created for Omiotis. Results in three TREC collections are encouraging and reveal that the proposed GVSM partially improves and constantly never deteriorates the VSM model. In the future, we plan to create a semantic indexing framework, so that semantic information from WordNet is indexed at index time of the documents, and not during their retrieval. This plan has many challenging points that remain to be solved, and thus it constitutes interesting future work. Once developed, it will allow on-line text retrieval using semantic information from word thesauri.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Chapter 6

Conclusions

This thesis has investigated the use of semantic information from word thesauri in several text applications like text classification, text retrieval, paraphrasing and word-to-word relatedness. The aim of this thesis was to provide new methods for extracting semantics from text that exist in a word thesaurus through state of the art Word Sense Disambiguation (WSD) approaches, and to propose novel models for embedding disambiguation information into the aforementioned applications. A novel measure of semantic relatedness based on WordNet was defined, Omiotis, which was experimentally shown to capture successfully the semantic relatedness between words and text segments, matching in several data sets the human performance in similarity perception between pieces of text. New models for embedding the Omiotis measure in challenging text applications, like text classification and text retrieval, were proposed, and we have shown that they perform better than traditional document similarity measures and models, like the combination of VSM and cosine similarity. We believe that the proposed solution has accomplished its initial goal, which was to find new means of computing similarity between text, besides relying solely on the exact keyword matching that many traditional models adopt. In this direction, this work constitutes a new methodology for processing text with the aim to classify and retrieve or organize document collections. In the following, we discuss the contribution of this work in detail, we sum up the conclusions reached after the experimental evaluation, and finally we provide useful directions towards the continuation of the

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

current research.

6.1 Contributions

The contributions of this research span in three directions. Primarily, new methods for Word Sense Disambiguation are introduced that have been shown to achieve state of the art performance in three benchmark WSD data sets. Secondly, a novel measure of semantic relatedness is introduced, that embeds the semantic information from WordNet into the measurement of word-to-word (SR measure) and text-to-text (Omiotis measure) relatedness. Finally, new models for several text applications are introduced that use SR and Omiotis.

WSD

We have introduced four new methods for WSD, three unsupervised and one supervised, all belonging to the category of knowledge-based WSD. The first approach (compactness-based WSD) exploits the content and structure (i.e., the senses and hierarchical relationships) of hierarchical thesauri (HT) and extends the bag of words model for text classification. The contribution of this approach is the design of a successful WSD method that improves the text classification process. The compactness WSD approach takes into account term senses found in HTs, (in the specific case Wordnet), and for each document selects the best combination of them based on their conceptual compactness in terms of related Steiner tree costs. Apart from the senses we add to the original document feature set a controlled number of hypernyms of the senses at hand. The hypernyms are incorporated by means of the GVSM kernel utilized.

The attractive features of this approach are:

(i) Appropriate WSD approach for text classification. Most of the related approaches incorporating WSD in the classification task do not provide a sound experimental evidence on the quality of their WSD approach. On the contrary in this approach, the WSD algorithm is exhaustively evaluated against various humanly disambiguated benchmark datasets and achieves very high precision (among the top

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

found in related work) although at low coverage values. The experimental evaluation provides us with the assurance that our WSD algorithm can be configured to have high precision, and thus, would insert in the training set very little noise.

(ii) Similarity measure that takes into account the structure of the HT. Document classification depends on a relevant similarity measure to classify a document into the closest of the available classes. It is obvious that the similarity among sets of features (representing documents) should take into account their hierarchical relationships as they are represented in the HT. None of the previous approaches for embedding WSD in classification has taken into account the existing literature for exploiting the HT relations. Even when hypernyms are used, they are used in an ad-hoc way, based on the argument that the expansion of a concept with hypernyms would behave similar to query expansion using more general concepts. We utilize a Kernel based on the general concept of a GVSM kernel that can be used for measuring the semantic similarity between two documents. The kernel is based on the use of hypernyms for the representation of concepts - theoretically justified by previous related work concerning the computation of semantic distances and similarities on a HT that aligns to tree structure. We conducted classification experiments on two real world data sets (the two largest Reuters categories and a data set constructed from the editorial reviews of products on three categories of the Amazon web site). The results demonstrate that this approach for embedding WSD in classification yields significantly better results, when the training sets are small.

In an effort to improve even more the disambiguation performance and close the gap between automated WSD and human performance (which in the used data sets ranges from 67% to 80%), we presented a new unsupervised WSD algorithm, which utilizes all types of semantic relations in WordNet. The algorithm uses Spreading Activation Networks (SANs), but unlike previous WSD work it creates SANs taking into account all sense-to-sense relations, rather than relations between senses and glosses, and it employs a novel edge-weighting scheme. The algorithm was evaluated on three benchmark data sets (Senseval 2, 3 and SemCor), using WordNet as the thesaurus, though it is general enough to exploit other word thesauri as well. It outperformed: (i) the most recent SAN-based WSD method, which overcame the

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

problems older approaches faced, and (ii) the best unsupervised WSD methods that participated in the respective Senseval competitions.

Though SANs provided an accuracy of around 50%, we explored the possibility of keeping the same, rich, semantic representation of the defined semantic networks but changing the processing algorithm of the nodes, using a variation of PageRank that takes into account weights on edges. This new WSD algorithm boosted the WSD performance by an additional $5 - 7\%$. Currently, to the best of our knowledge, the PageRank-based method is the top performing unsupervised knowledge-based WSD approach in the WSD bibliography [79], excluding ensembles.

Finally, we have introduced a new supervised, multilayered WSD method based on an ensemble of three WordNet-based WSD algorithms. The method trains a set of SVM classifiers, one for each WSD algorithm, and learns which WSD method to trust depending on the feature vector of the target term. The contributions of this method are: (i) state of the art accuracy in unrestricted text WSD, (ii) limited training requirements to achieve top performance, (iii) low space complexity, since the classifiers are trained on a very small number of features and the stored support vectors are on average 3% of the training instances, and (iv) the disambiguation step of the algorithm has low time complexity, since it is reduced to executing a sing base WSD method for each word occurrence, selected from a list of methods that do not require training (SANs, PR and FS). The proposed approach performs as well, in terms of accuracy, as state of the art methods for unrestricted text WSD, and has low space and execution time requirements for the disambiguation step.

Measuring Semantic Relatedness

Another contribution of this work is the formulation of Omiotis, a new measure of semantic relatedness for text segments. The major strength of Omiotis is the formulation of the semantic relatedness between words (SR), which is exploited to measure semantic relatedness between texts. Omiotis' innovation is that it combines for the first time three important factors: (i) the length of the path that connects the senses of words in the used thesaurus; (ii) the senses' depth in the thesaurus, and (iii) the importance of the thesaurus' edges. Furthermore, it uses all of the available

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

semantic information in the thesaurus, even semantic links that cross parts of speech, and this enables the measure to compute relatedness for pairs of words of every POS combination. Experimental evaluation showed that our measure approximates human judgements of semantic similarity between words better than previously proposed measures.

Applications

Omiotis can be embedded in challenging text applications. For text classification, we have created a new semantic smoothing kernel that embeds the semantic relatedness measure into the support vector machines learning mechanism. The semantic kernel outperforms both the linear kernel and the GVSM kernel that uses compactness disambiguation information in the Reuters data set. Results reveal that the semantic relatedness between text segments captured by our measure produce very high macro-F1 scores, especially when small training sets are used. In text retrieval, we have incorporated the semantic relatedness measure into a new GVSM that extends the naive VSM model. Experiments in three TREC collections show that the semantic information from WordNet improves in many cases the retrieval performance and never deteriorates it.

In addition, we have used our semantic relatedness measure in several other interesting applications. Primarily, we have incorporated the measure into a formula that can produce relatedness scores of candidate word pairs in SAT questions. Results show that our measure surpasses every other knowledge-based measure used in the task and matches the performance of several corpus-based measures that have high execution time (i.e. several hours for the set of the 374 examined SAT questions, while our measure needs less than one minute). Additionally, we have used Omiotis in the text paraphrase recognition task. Experimental evaluation shows that our measure reduces the error rate of the standard vectorial model by 14.69%, indicating that it captures successfully text relatedness.

Finally, the largest part of the research conducted for the writing of this thesis has been published in the following articles (chronological order):

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

- G. Tsatsaronis, I. Varlamis, M. Vazirgiannis and K. Nørvåg, "Omiotis: A Thesaurus-based Measure of Text Relatedness", Demo paper, to appear in the proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (PKDD 2009), Slovenia, 2009.

- G. Tsatsaronis and V. Panagiotopoulou. A generalized vector space model for text retrieval based on semantic relatedness. In Proc. of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009, Student Research Workshop), pages 70-78, 2009.

- G. Tsatsaronis, I. Varlamis, and M. Vazirgiannis. Word sense disambiguation with semantic networks. In Proc. of the 11th International Conference on Text, Speech and Dialogue (TSD 2008), pages 219-226, 2008.

- G. Tsatsaronis, M. Vazirgiannis, and I. Androutsopoulos. Word sense disambiguation with spreading activation networks generated from thesauri. In Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), pages 1725-1730, 2007.

- D. Mavroeidis, G. Tsatsaronis, and M. Vazirgiannis. Semantic Distances for Sets of Senses and Applications in Word Sense Disambiguation, Book Chapter in: Knowledge Mining, (Ed.): S. Sirmakessis. Springer Verlag, 2005.

- M. Eirinaki, D. Mavroeidis, G. Tsatsaronis, and M. Vazirgiannis. Introduc- ing Semantics in Web Personalization: The role of Ontologies. Book Chapter in: Semantics, Web, and Mining, (Ed.): M. Ackerman, B. Berendt, M. Grobelnik, A. Hotho, D. Mladenic, G. Semeraro, M. Spiliopoulou, G. Stumme, V. Svatek, M. van Someren. Springer Verlag, 2005.

- D. Mavroeidis, G. Tsatsaronis, M. Vazirgiannis, M. Theobald, and G. Weikum. Word sense disambiguation for exploiting hierarchical thesauri in text classification. In Proc. of the 9th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2005), pages 181-192, 2005.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

6.2 Conclusions

From the work conducted in this thesis, we have drawn several important conclusions regarding semantic information and its use in text applications. These can be summarized as follows:

- Automated WSD can reach human performance, even in fine-grained disambiguation text collections, like Senseval 2, 3 and SemCor disambiguated with WordNet 2.0, if all of the available semantic information from the thesaurus is used, and an ensemble of approaches is utilized.

- The semantic information that WordNet offers is rich, and there are means of embedding it into text applications that can improve the performance of traditional models, like VSM. We have introduced such models for several applications, like text classification, text retrieval, paraphrase recognition, and SAT analogy tests.

- Semantic relatedness between words and between text segments can be captured successfully by measures of semantic relatedness that take into account in tandem all factors affecting the connectivity of nodes in a semantic graph: semantic path length, the senses' depth, and the importance of edges.

- It is computationally feasible to incorporate of semantic information into text applications, but an infrastructure with pre-indexed sense-to-sense values of relatedness is needed.

6.3 Future Work

This work creates much space for new theoretical models that use semantic networks and semantic document representation, but also in the application level of text processing. In WSD the next research steps should concentrate on studying additional methods, both supervised and unsupervised, that can complete the introduced ensemble, to match the human performance in unrestricted text. Current state of the


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

art research in WSD focuses on a small improvement window of $1-5\%$ from the most frequent sense baseline, but improvements in this window may allow WSD methods to match human performance, and in this direction the ensemble approach seems very promising.

Regarding the research area of measures of semantic similarity and relatedness, future work should concentrate on capitalizing on the knowledge that publicly available corpora can offer, when information from thesauri does not suffice. For example, current trends include, among others, the use of Wikipedia as an alternative knowledge base to thesauri like WordNet.

Furthermore, we plan to apply our semantic relatedness measure to more applications, such as text clustering, keyword and sentence extraction (using SemanticRank, the algorithm we propose for keyword extraction), query expansion etc. and to examine how the measure can be tuned further to achieve better performance.

Moreover, regarding the research in GVSM models for text retrieval, there are additional aspects that deserve further attention. In some previously proposed GVSMs, it was suggested that semantic information can create an individual space, leading to a dual representation of each document, namely, a vector with document terms and another one with semantic information. Similarly, our proposed GVSM could complement the standard VSM representation. Thus, the similarity between a query and a document may be computed by weighting the similarity in the terms space and the senses space.

Finally, an interesting aspect of the use of semantic information in text applications is the creation of an indexing infrastructure that will not only index terms existing in every incoming document, but will store semantically related concepts during the indexing process as well. This will allow embedding of semantic information even in real time applications. This is also the imminent research interest of the author.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Appendix A

WordNet 2.0 Structure

Traditionally, machine readable dictionaries (MRD), like the Collins English dictionary, were used in text related tasks. (i.e early attempts on word sense disambiguation). Word thesauri like WordNet [31], or Roget's International Thesaurus [76], constitute the knowledge-base for several text-related research tasks. WordNet is the knowledge base used in this thesis. WordNet's lexical database contains English nouns, verbs, adjectives and adverbs, organized in synonym sets (synsets). The terms *senses* and *synsets* are used interchangeably the thesis. Synsets are connected with various edges, representing semantic relations among them, and the latest WordNet versions, like 2.0, offer a rich set of such links: hypernymy / hyponymy, meronymy / holonymy, synonymy / antonymy, entailment / causality, troponymy, domain / domain terms, derivationally related forms, coordinate terms, attributes, and stem adjectives. As it is shown in figure A.1, several relations cross parts of speech, like the *domain terms* relation, which connects senses pertaining to the same domain (e.g. *light*, as a noun meaning electromagnetic radiation producing a visual sensation, belongs to the domain of *physics*). In all cases, when WordNet 2.0 is used in any of the proposed approaches, all of the offered semantic relations existing in WordNet 2.0 are utilized.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

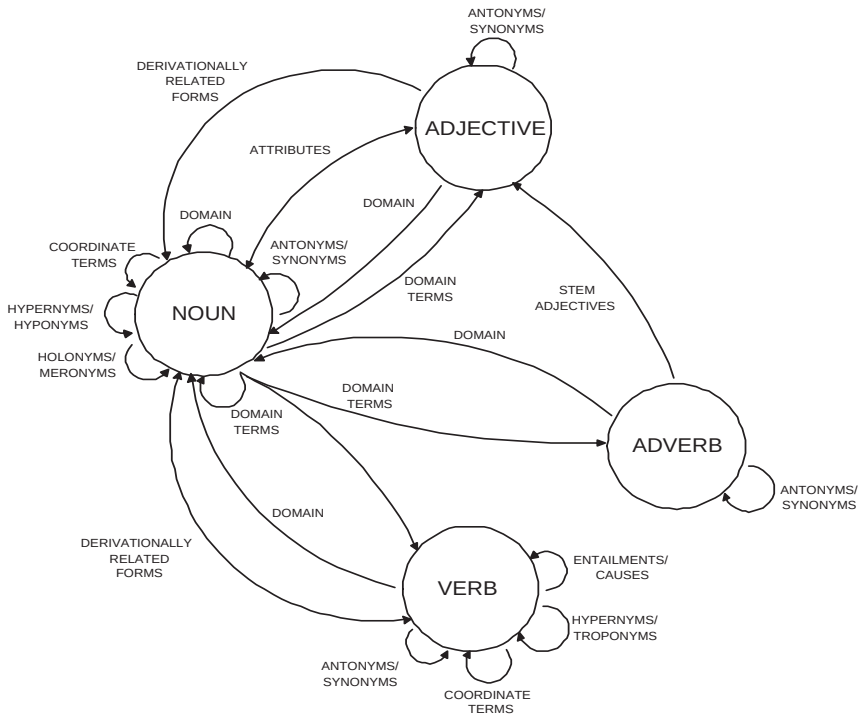

Figure A.1: Semantic relations in WordNet 2.0.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Appendix B

Complexity and Implementation

The computation of Omiotis entails a series of steps, the complexity of which is strongly related to its base measure of Semantic Relatedness (SR). Primarily, a fast API to WordNet is needed, in order to retrieve semantic information lying in its lexical database. In order to access WordNet fast, we make use of an open source library written in Java, namely Java WordNet library[1]. This Java API to WordNet provides easy access to the semantic information in WordNet's lexical database. On top of that API we have developed a wrapper (JWNLWrapper) that handles all the basic functions needed from the algorithms described in this thesis. More details on the used API and the developed wrapper are provided in appendix B.4.

Furthermore, a fast implementation of algorithm 4 is crucial to the overall performance, since it entails the computation of the path that maximizes a certain product. Finally, integration with existing retrieval platforms for performing fast text retrieval using Omiotis is also needed. In this chapter we describe the overall system implementation and the infrastructure that we have created in order to make Omiotis and SR scalable measures that can handle large amounts of data.

[1] `http://sourceforge.net/projects/jwordnet`

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

B.1 Complexity Issues

Primarily, given two words, w_1 and w_2 the construction time of the semantic network used to compute SR according to algorithm 4, we proved in [121] to be $O(2 \cdot k^{l+1})$, where k is the maximum branching factor of the used thesaurus nodes and l is the maximum semantic path length in the thesaurus. Once the semantic network is constructed, the complexity of algorithm 4 is reduced to the standard time complexity cost of Dijkstra's algorithm. Using Fibonacci heaps, it is possible to alleviate the computational burden of Dijkstra and further improve time complexity. In the semantic network, Dijkstra takes $O(nL + mD + nE)$, where n is the number of nodes in the network, m the number of edges, L is the time for insert, D the time for decrease-key and E the time for extract-min. If Fibonacci heaps are used then $L = D = O(1)$ and the cost of extract-min is $O(logn)$, thus significantly reducing the cost of execution. This whole procedure is repeated $2 \times n_1 \times n_2$ times for the computation of Omiotis between two documents d_1 and d_2 having in total n_1 and n_2 distinct words respectively. More details on the use of Fibonacci heap with the Dijkstra algorithm can be found in appendix C.

B.2 Omiotis Implementation

From the aforementioned, it is obvious that the computation of Omiotis is not cheap in general. For this purpose, and in order to improve the system's scalability, we have pre-computed and stored all SR values between every possible pair of synsets in a RDBMS. This is a one-time computation cost which dramatically decreases the computational complexity of Omiotis. The database schema has three entities, namely *Node*, *Edge* and *Paths*. *Node* contains all WordNet synsets. *Edge* indexes all edges of the WordNet graph adding weight information for each edge computed using the SR measure. Finally, *Paths* contains all pairs of WordNet synsets that are directly or indirectly connected in the WordNet graph and the computed relatedness. These pairs were found by running a Breadth First Search (BFS) starting from all WordNet roots for all POS. Table B.1 provides statistical information for the RDBMS

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Synsets	Edges	Con. Synset Pairs	Avg In-Degree	Avg Out-Degree	Avg Fan-In	Avg Fan-Out
110,490	324,268	11,182,324,723	2.9933	2.9535	103,192.32	101,822.56

Table B.1: Statistics of the WordNet 2.0 graph in the implemented database.

which exceeds 300 Gbytes in size. Numbers in columns 4 and 5 measure the average in- and out-degree based on direct edges between synsets and numbers reveal that WordNet graph is asymmetric, which is due to the Stem Adjectives and Derivational Related Forms relations, which direct from a synset s_i to synset s_j but not the opposite. Numbers in columns 6 and 7 reveal the same when considering all possible paths between any synset s_i and synset s_j. This is again due to the aforementioned asymmetry. The current implementation takes advantage of the database structures (indices, stored procedures etc) in order to decrease the computational complexity of Omiotis. The following example is indicative of the complexity of SR computation. The average number of senses per term is between 5 and 7 (depending on the POS). For a pair of terms of known POS, we perform $\frac{n^2}{2}$ ($n \simeq 6$) combinations and for each pair of synsets we compute the similarity as presented in definition 5. When these similarities are pre-computed, the time required for processing 100 pairs of terms is $\simeq 1$ sec, which makes the computation of Omiotis feasible and scalable. As a proof of concept, we have developed an on-line version of the SR and the Omiotis measures[2], where the user can test the term-to-term and sentence-to-sentence semantic relatedness measures and the respective execution times.

B.3 Integration with Terrier

Terrier (Terabyte Retriever)[3] is an open source retrieval platform which implements a wide variety of term weighting and retrieval models. It is focused on the TREC collections[4], though it is able to index other collections as well. Since in our evaluation we experiment with several TREC collections, we have integrated the infrastructure

[2]Publicly available at `http://83.212.240.72`
[3]`http://ir.dcs.gla.ac.uk/terrier/`
[4]`http://trec.nist.gov/tracks.html`

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

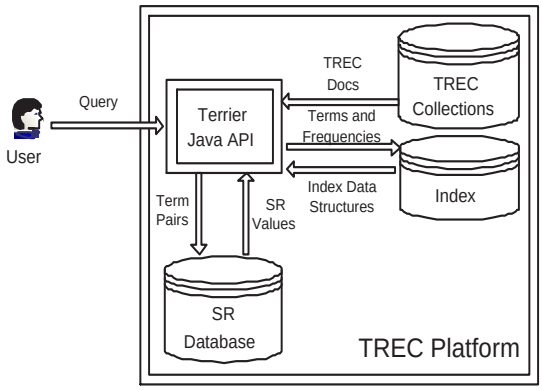

Figure B.1: SR Integration with Terrier Platform.

described in the previous section into the TREC architecture. In figure B.3 an abstract overview of the integration is shown. The integration consists of manipulating the component of the Terrier Java API that handles the similarity of a given query with the document. When a query is submitted, prior to returning the results, we incorporate the semantic relatedness between every query term and every document term inside the similarity function, in order to compute the value of the GVSM explained in the previous chapter. Essentially, and because this procedure is conducted on-line, the time that the system needs to respond, increases dramatically. But, as more queries are submitted to the system, and because we also index the computed SR values among terms, the systems' execution time performance improves. Alternatively, we need to pre-compute all the pairwise SR values for all lemmas existing in WordNet (as we have done in the sense level). This procedure is in our next plans to implement, and there is evidence that it will improve the response time even more. Currently, and for the purposes of our experimental evaluation in three TREC collections, we have pre-computed the SR values between query terms and the top 50 retrieved documents from the VSM in the normal Terrier mode. This allows us to evaluate the performance of SR in the returned top 50 results of the VSM model. Expansion of the evaluation in more top documents (e.g. top 1000) is feasible, but the computational cost is still high, unless we expand the SR database with precomputed term SR values between lemmas in WordNet.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

B.4 Examples of Accessing WordNet with the JWNL Wrapper

For accessing WordNet, we have used the Java WordNet library, a free and open source Java API for communicating with the index files of WordNet's lexical database. In order to provide more functionality to the JWNL API, we have created a wrapper that provides the basic functionality needed for implementing the algorithms presented in this thesis. Below follow some examples of accessing WordNet by using the JWNL wrapper[5].

Initially the JWNLWrapper initializes the JWNL API for the basic interaction with the physical files of WordNet (WordNet uses a separate large index for each POS). This is done as shown next.

```
JWNL.initialize(new FileInputStream("conf\\file_properties.xml"));
```

Then, given that a JWNLWrapper object is created (e.g. wnWrapper), we can use this object to retrieve all the WordNet senses for a given lemma, like shown in the following command. The command will return an ArrayList of Synset objects (defined in JWNL) for the lemma *Test*. Each Synset object is essentially a concept of WordNet thesaurus. In this case, it will return 6 synsets for the noun *test* and 7 synsets for the verb *test*.

```
ArrayList<Synset> lemmaSenses = wnWrapper.getSenses(String lemma);
```

Then, the synsets can be processed with the functionality that the JWNL API provides (e.g. get the POS, the gloss words, the synsets connected to it, etc.).

Basic Utility Functions

In order to improve the functionality of the JWNL API, we have developed in the JWNLWrapper a series of methods that help us process WordNet faster and easier. The signatures of some of those methods are shown in the following. The methods names are self-explanatory. Note that the PointerTargetTreeNode is connected to a Synset object, but stores more information regarding the specific WordNet node.

[5]Available for download from `http://www.db-net.aueb.gr/gbt/download.html`


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
public void printChildren(Synset node, int depth);

public ArrayList<Synset> getHyponyms(PointerTargetTreeNode node, int depth);

public ArrayList<Synset> getAllWordnetSensesOfPartOfSpeech(String lemma, String POS);

public int getDepthOfSynset(Synset aSynset);

public ArrayList<Synset> getHypernyms(PointerTargetTreeNode node);

public ArrayList<Synset> getConnectedSynsets(Synset aSynset);
```

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Appendix C

Dijkstra Using Fibonacci Heaps

Algorithm 4 that computes the semantic relatedness between a pair of synsets is the modified Dijkstra algorithm to maximize the product explained in definition 5, with an appropriate weighting of the edges. Note however that in the literature [23] the reference to the *modified Dijkstra algorithm* is normally used for the version of Dijkstra that uses a binary heap to implement the priority queue Q of the Dijkstra algorithm. But, as we will explain in the following, we have used Fibonacci Heaps, instead of the binary heaps.

The original Dijkstra algorithm has a running time of $O(V^2 + E) = O(V^2)$, where V is the set of vertices of the graph and E the set of edges. In the case of sparse graphs, like the constructed semantic networks from WordNet in our case, if binary heaps are used, the total running time is reduced to $O((V+E)logV)$. Fibonacci heaps can obtain a further reduction of the running time, achieving $O(VlogV+E)$ and since $|V|$ is usually several hundreds or even thousands in our constructed networks, this implementation produces an even faster solution.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Appendix D

Effect of Lexical Ambiguity in Five Toy IR Data Sets

In the field of IR much work has been done to estimate the effect of each lexical ambiguity type independently in retrieval performance. In an effort to provide such an analysis, the current appendix procvides: (a) experimental study on the effect of each ambiguity type in IR, (b) evaluation of three GVSM, and (c) analysis on the intrinsic features of query and documents words whose disambiguation may improve, but not deteriorate IR performance. Among the related studies made in the past, a special focus must be given to the works by Krovetz and Croft [51], Voorhees [133], Sanderson and van Rijsbergen [102], and Stokoe [111]. Besides Voorhees, who used a real WSD system, the rest used a methodology based on the generation of random pseudowords to assess the role of semantic ambiguity in IR. This appendix is a complementary experimental work to the aforementioned in several aspects. Primarily, it investigates the relation of each ambiguity type with IR independently. Also, it tests 3 GVSM models and highlights the features of words that should be disambiguated, so that WSD can improve IR performance. Finally, it uses WordNet and a simple WSD system (most frequent sense given by WordNet for each word occurrence), instead of pseudowords, to assess sense ambiguity in IR.

The representation of documents and queries in all tested retrieval models is based on the simple baseline vector space model (VSM) with a conventional ranking system

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

of $TF \cdot IDF$ as a weighting scheme for all terms [5]. Similarity between query and document vectors in all models is computed with the *cosine* measure. For every discussed type of ambiguity, we altered the VSM representation of documents and queries, and created a new retrieval model for each ambiguity resolution. For the syntactic ambiguity, we considered a document space where each term is indexed along with its POS. For each term occurrence t_i in a document (the same holds for queries), we index the pair $t_i\_POS_i$, where POS_i is its POS for this occurrence. POS is found using the *Stanford Log-Linear POS Tagger*. For sense ambiguity, we created 3 different GVSM, which embed senses information. All queries were manually disambiguated and a baseline WSD system is applied to the documents (selecting always the first sense from WordNet). Along with the terms, we added in the vectors the senses that disambiguate them. The difference in the 3 GVSM lies in the weighting of terms and senses. GVSM1 considers two different vector spaces, terms and senses. Each one uses its own, separate $TF \cdot IDF$ weighting, and the final similarity between a document and a query is computed as the sum of the cosine similarities in the two spaces. GVSM2 considers terms and senses as dimensions in the same space. Thus, a new hybrid vector space, with each distinct term and sense being dimension, is created. GVSM3, that we proposed in [63], considers one hybrid vector space of terms and senses, though senses are assigned with the $TF \cdot IDF$ weight of their terms. When two distinct terms are disambiguated with the same sense, the terms' weights are combined to produce the sense weight. For the resolution of phrases' ambiguity we perform a phrase detection algorithm using a sliding window of varying length, starting from 7 terms and dropping down to 2. Phrases are recognized with dictionary look-up in WordNet. The resulting phrases substituted the respective term occurrences in the indexing. Finally, the effect of stemming is studied with the use of Porter stemmer. The stems substituted the respective term occurrences in the indexing.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

	D.	Q.	Domain	#T D.	#T Q.	#P D.		#P Q.		D. amb.		Q. amb.	
CACM	3204	64	ACM abstr.	29.5	13.4	2.11	49.7%	1.4	39%	5	88.3%	4.5	84.9%
MED.	1033	30	med. abstr.	82.3	11.6	2.9	89%	1.3	66.6%	4.4	82.2%	3.1	90.3%
TIMES	423	83	general	304.6	8.2	9.8	100%	1.3	57.8%	4.5	70.5%	3.6	77.9%
NPL	11429	100	physics	23.4	6.8	1.4	44.9%	1.2	26%	4.6	70.2%	4.3	62.1%
CRAN.	1400	225	aer/mics	90.4	9.3	3.1	88%	1.1	35.1%	5.3	88.1%	4.7	89.9%

Table D.1: Documents, queries and domains of the retrieval collections.

D.1 Description of the Data Collections

Experimental evaluation was conducted in 5 IR collections [1], namely CACM, CRANFIELD, NPL, MEDLINE and TIMES. Table D.1 presents the details of the used data sets. The domain of each collection, the number of documents (D.) and queries (Q.) and the average number of term occurrences in the documents (#T D.) and the queries (#T Q.)are shown. Also, table D.1 shows the average number of phrases recognized in the documents (#P D.) and the queries (#P Q.). Note, however, that these numbers refer to the documents and queries in which at least one phrase was found. In the same columns we also report the percentages of documents and queries that at least one phrase was recognized from the dictionary. Finally, we report the average ambiguity of words found in WordNet for all documents (D. amb.) and queries (Q. amb.) per collection, which is the average number of WordNet senses for these words, along with the percentage of terms found in the used lexicon, separately for queries and documents.

D.2 Results and Analysis

Figure D.1 presents per collection the differences of each retrieval model from the interpolated precision of the VSM baseline (indexed terms in the collection without resolving any ambiguity) for the 11-standard recall points. The depicted lines are produced from the execution of the six discussed retrieval models, namely *stemming*

[1]Material used is downloadable from http://www.db-net.aueb.gr/gbt/download.html

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

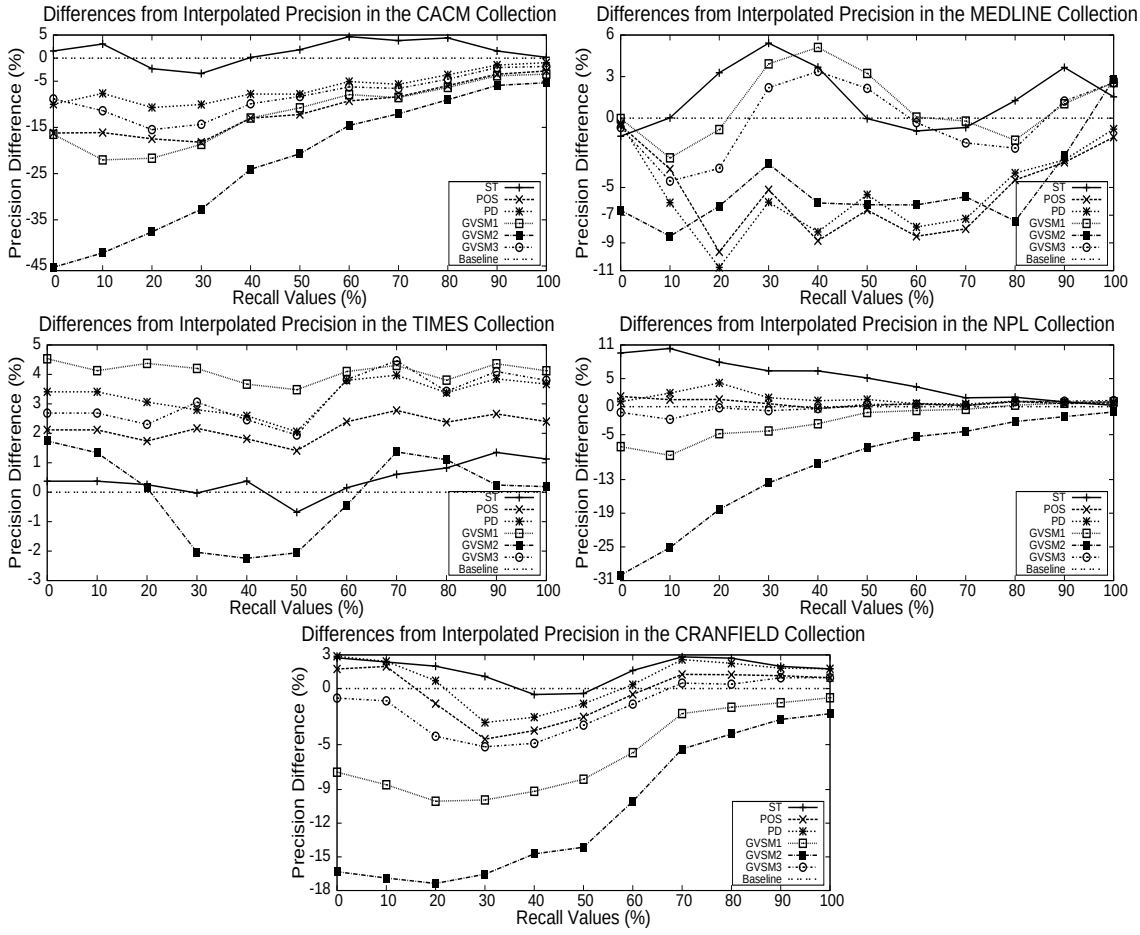

Figure D.1: Differences from the baseline in interpolated precision.

(ST), namely *POS tagging* (POS), *phrase detection* (PD), and the three GVSM models that handle sense ambiguity (GVSM1, GVSM2 and GVSM3). Below follows the analysis for each retrieval model separately.

Stemming: As shown in figure D.1, stemming improves almost in all cases IR performance. In four collections (CACM, MEDLINE, TIMES and CRANFIELD) stemming produces the same behavior in the respective retrieval model. Initially it boosts IR precision, compared to the baseline, for the first 2 or 3 recall points, then it weakens in the $40\% - 60\%$ recall levels, and finally it boosts precision again, up to the point where all relevant documents are retrieved. The only collection where stemming boosted constantly and high (adding up to 11% in precision), without drop

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

in any recall point, is NPL. Observing closely NPL from table D.1, we notice that it has the shortest documents and queries among all used collections. By examining further the boost in precision due to stemming, we notice that the highest differences occur for the NPL, CACM and MEDLINE (in all cases more than 4.5% for certain recall points), which also happen to be the collections with the shortest documents. Thus, according to our results we conclude that stemming can boost precision in low and high recall levels, while it boosts more in cases when documents are relatively small.

POS Tagging: Resolving POS in IR cannot help much in precision, as shown in figure D.1. The POS retrieval model can help as much as 2.5%, compared to the baseline, and this only happens in TIMES collection. In the CRANFIELD collection it boosts precision by 2%, but in the same collection also drops up to 5% in the medium recall points. Note also that in CACM, it drops by even 18%. Examining further TIMES and CACM collections, in which POS model achieved its top and its worst performance respectively, we notice that TIMES has the largest documents and CACM is among the two collections with the shortest. From our analysis, it can be inferred that POS information cannot boost precision dramatically, but in the cases it boosts at all, the collections have relatively large documents.

Phrase Detection: Phrase detection can boost retrieval performance up to 4%, as shown in figure D.1. Looking closer at the collections where PR improves precision, we notice that this happens only with TIMES and NPL, where it boosts precision in all recall levels. Reversely, in CACM, the effect of phrase detection is negative, as it drops precision by almost 10% in the first 3 recall points. A close comparative examination of TIMES and CACM from table D.1, reveals that TIMES has many phrases detected on average per document (9.8), while CACM very few (2.1). Furthermore, in TIMES at least one phrase was detected in all documents, while in CACM this only happened in 39% of the documents. Finally, at least one phrase was recognized in a larger percentage of documents in TIMES, than in CACM. Overall, these findings lead us to the conclusion that phrase detection needs large coverage to aid IR performance, both in documents and queries.

WSD: The 3 GVSM used show that embedding WSD information in a retrieval

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

model can boost precision up to 5%, but can also drop it by more than 45%. A close look at figure D.1 reveals that GVSM1 and GVSM3 have very similar behavior, and both are better from GVSM2. The latter can only improve precision by at most 2% (TIMES), while it can drop very much performance (CACM). In general, considering a single vector space and mixing up term and sense dimensions, treating them as even dimensions in a VSM, seems a bad choice. In contrast, handling sense and terms in separate vector spaces (GVSM1), or weighting senses with their terms' weights (GVSM3) can produce a boost in IR performance (TIMES, MEDLINE, NPL). Observing closely TIMES and MEDLINE, where GVSM1 and GVSM3 can help IR noticeably, we discover that precision boost is between $2-5\%$. In TIMES, this takes place in all recall points. In MEDLINE, this occurs for the larger recall points. The two collections have a remarkable property. They are the collections with the less ambiguous terms, both per document and per query, as we see from table D.1. Our results allow us to conclude that a WSD system with an average accuracy of around 60% (this is the first sense heuristic performance in Senseval-2 and Senseval-3 WSD data sets as discussed in chapter 3) can boost retrieval performance, by even 5% in cases where the average ambiguity of the disambiguated terms is relatively low (i.e., maximum 4 senses per term).

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Bibliography

[1] E. Agirre and P. Edmonds. *Word Sense Disambiguation Algorithms and Applications*. Springer, 2007.

[2] E. Agirre and G. Rigau. A proposal for word sense disambiguation using conceptual distance. In *Proc. of the 1st International Conference on Recent Advances in NLP*, 1995.

[3] E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 16–22. Copenhagen, Denmark, 1996.

[4] E. Altintas, E. Karsligil, and V. Coskun. The effect of windowing in word sense disambiguation. In *Proc. of the 20th International Symposium on Computer and Information Sciences*, pages 626–635. Springer Verlag, 2005.

[5] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[6] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proc. of IJCAI 2003*, 2003.

[7] R. Basili, M. Cammisa, and A. Moschitti. A semantic kernel to exploit linguistic knowledge. In *Proc. of the AI*IA 2005*, pages 290–302, 2005.

[8] H. Berger, M. Dittenbach, and D. Merkl. An adaptive information retrieval system based on associative networks. In *Proc. of the 1st Asia-Pacific Conference on Conceptual Modelling*, pages 27–36, 2004.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[9] E. Bicici and D. Yuret. Clustering word pairs to answer analogy questions. In *Proc. of the Fifteenth Turkish Symposium on Artificial Intelligence and Neural Networks*, 2006.

[10] S. Bloehdorn, R. Basili, M. Cammisa, and A. Moschitti. Semantic kernels for text classification based on topological measures of feature similarity. In *Proc. of the ICDM 2006*, pages 18–22, 2006.

[11] S. Bloehdorn and A. Hotho. Boosting for text classification with semantic features. In *Proc. of the SIGKDD 2004, Mining for and from the Semantic Web Workshop*, 2004.

[12] S. Bloehdorn and A. Moschitti. Exploiting structure and semantics for expressive text kernels. In *Proc. of the CIKM 2007*, pages 861–864, 2007.

[13] B. Boguraev and J. Pustejovsky. Lexical ambiguity and the role of knowledge representation in lexicon design. In *Proc. of the International Conference on Computational Linguistics.*, pages 36–41, 1990.

[14] D. Bollegala, Y. Matsuo, and M. Ishizuka. Www sits the sat: Measuring relational similarity from the web. In *Proc. of ECAI 2008*, pages 333–337, 2008.

[15] L. Bookman. A microfeature based scheme for modelling semantics. In *Proc. of IJCAI 1987.*, pages 611–614, 1987.

[16] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:1–7, 1998.

[17] S. Brody, R. Navigli, and M. Lapata. Ensemble methods for unsupervised wsd. In *Proceedings of COLING/ACL 2006*, pages 97–104. Sydney, 2006.

[18] A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[19] S. Chakrabarti. *Mining the Web.* Morgan Kaufmann, 2003.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[20] T. Chklovski and R. Mihalcea. Exploiting agreement and disagreement of human annotators for word sense disambiguation. In *Proc. of the Conference on Recent Advances on Natural Language Processing (RANLP)*, 2003.

[21] R.L. Cilibrasi and M.B. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, 2007.

[22] P. Clough and M. Stevenson. Cross-language information retrieval using eurowordnet and word sense disambiguation. In *Proc. of the 26th ECIR*, pages 327–337, 2004.

[23] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. The MIT Press, 1990.

[24] G. Cottrell and S. Small. A connectionist scheme for modelling word sense disambiguation. *Cognition and Brain Theory*, 6:89–120, 1983.

[25] J. Cowie, J. Guthrie, and L. Guthrie. Lexical disambiguation using simulated annealing. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 359–365. Nantes, France, 1992.

[26] F. Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11:453–482, 1997.

[27] B.V. Dasarathy. *Nearest Neighbor: Pattern Classification Techniques (Nn Norms : Nn Pattern Classification Techniques)*. IEEE Computer Society, 1990.

[28] A. Devitt and K. Vogel. The topology of wordnet: Some metrics.. In *Proc. of GWC 2004*, 2004.

[29] W.B. Dolan, C. Quirk, and C. Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proc. of the 20th COLING*, 2004.

[30] M. Eirinaki, D. Mavroeidis, G. Tsatsaronis, and M. Vazirgiannis. *Introducing Semantics in Web Personalization: The role of Ontologies. In: Semantics,*

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Web, and Mining, (Ed.): M. Ackerman, B. Berendt, M. Grobelnik, A. Hotho, D. Mladenic, G. Semeraro, M. Spiliopoulou, G. Stumme, V. Svatek, M. van Someren. Springer Verlag, 2005.

[31] C. Fellbaum. *WordNet – an electronic lexical database*. MIT Press, 1998.

[32] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. *ACM TOIS*, 20(1):116–131, 2002.

[33] R. Florian, S. Cucerzan, C. Schafer, and D. Yarowsky. Combining classifiers for word sense disambiguation. *Natural Language Engineering*, 8(4):327–341, 2002.

[34] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. of the 20th IJCAI*, pages 1606–1611. Hyderabad, India, 2007.

[35] W. Gale, K. Church, and D. Yarowsky. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proc. of the ACL 1992*, pages 249–256, 1992.

[36] D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983.

[37] M. Halliday and R. Hasan. *Cohesion in English*. Longman, 1976.

[38] T. Hirao, T. Fukusima, M. Okumura, C. Nobata, and H. Nanba. Corpus and evaluation measures for multiple document summarization with multiple sources. In *Proc. of COLING 2004*, pages 535–541, 2005.

[39] G. Hirst and D. St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In *WordNet: An Electronic Lexical Database, chapter 13*, pages 305–332, Cambridge, 1998. The MIT Press.

[40] V. Hoste, W. Daelemans, I. Hendrickx, and A. van den Bosch. Evaluating the results of a memory-based word-expert approach to unrestricted word sense

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

disambiguation. In *Proc. of the ACL Workshop on Word Sense Disambiguation*, 2002.

[41] T. Hughes and D. Ramage. Lexical semantic relatedness with random graph walks. In *Proc. of EMNLP 2007*, 2007.

[42] N.M. Ide and J. Veronis. Word sense disambiguation: the state of the art. *Computational Linguistics*, 24(1):1–40, 1998.

[43] A. Islam and D. Inkpen. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Trans. Knowl. Discov. Data*, 2(2):1–25, 2008.

[44] M. Jarmasz and S. Szpakowicz. Roget's thesaurus and semantic similarity. In *Proc. of Conference on Recent Advances in Natural Language Processing*, pages 212–219, 2003.

[45] J.J. Jiang and D.W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of ROCLING X*, pages 19–33, 1997.

[46] T. Joachims. *Making large-scale SVM learning practical. Advances in Kernel methods - support vector learning.* B. Scholkopf, C. Burges and A. Smola (ed.), MIT-Press, 1999.

[47] G.H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proc. of the Eleventh Conference on Uncertainty in Artificial Intelligence*. 338–345, 1995.

[48] D. Klein, K. Toutanova, H.T. Ilhan, S.D. Kamvar, and Manning C.D. Combining heterogeneous classifiers for word-sense disambiguation. In *Proc. of the SIGLEX/SENSEVAL Workshop on Word Sense Disambiguation*, pages 74–80, 2002.

[49] U.S. Kohomban and W.S. Lee. Learning semantic classes for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 34–41. Michigan, 2005.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[50] H. Kozima and T. Furugori. Similarity between words computed by spreading activation on an english dictionary. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics*, pages 232–239. Utrecht, The Netherlands, April 1993.

[51] R. Krovetz and W.B. Croft. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10(2):115–141, 1992.

[52] T.K. Landauer, P. Foltz, and D. Laham. Introduction to latent semantc analysis. *Discourse Processes*, 25:259–284, 1998.

[53] C.A. Le, V.N. Huynth, A. Shimazu, and Y. Nakamori. Combining classifiers for word sense disambiguation based on dempster-shafer theory and owa operators. *Data and Knowledge Engineering*, 63(2):381–396, 2007.

[54] C.A. Le, A. Shimazu, and V.N. Huynth. Word sense disambiguation by combining classifiers with an adaptive selection of context representation. *Journal of Natural Language Engineering*, 13(1):75–95, 2006.

[55] C. Leacock, G. Miller, and M. Chodorow. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–165, March 1998.

[56] M.D. Lee and D.J Navarro. Extending the alcove model of category learning to featural stimulus domains. *Psychonomic Bulletin and Review*, 9(1):43–58, 2002.

[57] M.D. Lee, B. Pincombe, and M. Welsh. An empirical evaluation of models of text document similarity. In *Proc. of the CogSci2005*, pages 1254–1259, 2005.

[58] M. Lesk. Automated sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proc. of the SIGDOC Conference*, pages 24–26, 1986.

[59] D. Lin. An information-theoretic definition of similarity. In *Proc. of the 15th International Conference on Machine Learning*, pages 296–304, 1998.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[60] K. Litkowski. Use of machine readable dictionaries for word-sense disambiguation in senseval-2. In *Proc. of Senseval-2*, pages 107–110, 2001.

[61] C.D. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press, 2000.

[62] D. Mavroeidis, G. Tsatsaronis, and M. Vazirgiannis. *Semantic Distances for Sets of Senses and Applications in Word Sense Disambiguation, In: Knowledge Mining, (Ed.): S. Sirmakessis*. Springer Verlag, 2005.

[63] D. Mavroeidis, G. Tsatsaronis, M. Vazirgiannis, M. Theobald, and G. Weikum. Word sense disambiguation for exploiting hierarchical thesauri in text classification. In *Proc. of the 9th PKDD*, pages 181–192, 2005.

[64] M.L. Mc Hale. A comparison of wordnet and roget's taxonomy for measuring semantic similarity. In *Proc. of COLING/ACL 1998 Workshop on Usage of WordNet in Natural Language Processing Systems*, 1998.

[65] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 280–287. Spain, 2004.

[66] R. Mihalcea, C. Corley, and C. Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proc. of the 21st AAAI*, 2006.

[67] R. Mihalcea and A. Csomai. Senselearner: Word sense disambiguation for all words in unrestricted text. In *Proc. of the ACL 2005 on Interactive poster and demonstartion sessions*, pages 53–56, 2005.

[68] R. Mihalcea and D. Moldovan. A method for word sense disambiguation of unrestricted text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 152–158. College PArk, Maryland, 1999.

[69] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 404–411, Spain, 2004.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[70] R. Mihalcea, P. Tarau, and E. Figa. Pagerank on semantic networks with application to word sense disambiguation. In *Proc. of the 20th International Conference on Computational Linguistics*, Switzerland, 2004.

[71] G. Miller, C. Leacock, R. Tengi, and T. Bunker. A semantic concordance. In *Proceedings of ARPA Workshop on Human Language Technology*, pages 303–308. Plainsboro, New Jersey, 1993.

[72] G.A. Miller and W.G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.

[73] D. Moldovan and V. Rus. Logic form transformation of wordnet and its applicability to question answering. In *Proc. of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 394–401, France, 2001. Association for Computational Linguistics.

[74] A. Molina, F. Pla, and E. Segarra. A hidden markov model approach to word sense disambiguation. In *Proc. of the 8th Iberoamerican Conference on Artificial Intelligence*, 2002.

[75] A. Montoyo, A. Suarez, G. Rigau, and M. Palomar. Combining knowledge- and corpus-based word-sense-disambiguation methods. *Journal of Artificial Intelligence Research*, 23:299–330, March 2005.

[76] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17:21–48, 1991.

[77] R. Navigli. Online word sense disambiguation with structural semantic interconnections. In *Proc. of the 11th EACL*, 2006.

[78] R. Navigli. A structural approach to the automatic adjudication of word sense disagreements. *Natural Language Engineering*, 14(4):547–573, 2008.

[79] R. Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2), Article 10, 2009.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[80] R. Navigli, K. Litkowski, and O. Hargraves. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proc. of the 4th InternationalWorkshop on Semantic Evaluations (SemEval)*, pages 30–35, 2007.

[81] R. Navigli and P. Velardi. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1075–1086, 2005.

[82] M. Palmer, H. Dang, and C. Fellbaum. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Journal of Natural Language Engineering*, 13(2):137–163, 2007.

[83] M. Palmer, C. Fellbaum, and S. Cotton. English tasks: All-words and verb lexical sample. In *Proceedings of Senseval-2*, pages 21–24. Toulouse, France, 2001.

[84] M. Pasca. Open-domain question answering from large text collections. In *CSLI Studies in Computational Linguistics. CSLI Publications, Distributed by the University of Chicago Press*, 2003.

[85] M. Pasca. Mining paraphrases from self-anchored web sentence fragments. In *Proc. of PKDD 2005*, pages 193–204, 2005.

[86] S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proc. of the 4th International Conference on Inbtelligent Text Processing and Computational Linguistics*, pages 241–257. Springer Verlag, 2003.

[87] T. Pedersen. A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 63–69. Seattle, Washington, 2000.

[88] S.P. Ponzetto and M. Strube. Knowledge derived from wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research*, 30:181–212, October 2007.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[89] R.M. Quilian. The teachable language comprehender: a simulation program and theory of language. *Communications of ACM*, 12(8):459–476, 1969.

[90] R.M. Quillian. The teachable language comprehender: a simulation program and theory of language. *Communications of ACM*, 12(8):459–476, 1969.

[91] P. Resnik. Using information content to evaluate semantic similarity. In *Proc. of the 14th IJCAI*, pages 448–453, Canada, 1995.

[92] P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[93] G. Rigau, J. Atserias, and E. Agirre. Combining unsupervised lexical knowledge methods for word sense disambiguation. In *Proceedings of the 35th annual meeting of the Association of Computational Linguistics*, pages 48–55. Madrid, Spain, 1997.

[94] C. Rocha, D. Schwabe, and M. Poggi de aAragao. A hybrid approach for searching in the semantic web. In *Proc. of WWW 2004*, pages 374–383, 2004.

[95] P. Rosso, E. Ferretti, D. Jimenez, and V. Vidal. Text categorization and information retrieval using wordnet senses. In *Proc. of the GWC 2004*, 2004.

[96] H. Rubenstein and J.B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

[97] G. Salton. *The SMART Retrieval System*. Prentice Hall, 1971.

[98] G. Salton and M.E. Lesk. Computer evaluation of indexing and text processing. *Journal of the ACM*, 15(1):8–36, 1968.

[99] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[100] M. Sanderson. Word sense disambiguation and information retrieval. In *Proc. of the 17th SIGIR*, pages 142–151, Ireland, 1994. ACM.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[101] M. Sanderson. Ambiguous queries: Test collections need more sense. In *Proc. of the 31st SIGIR*, pages 499–506, 2008.

[102] M. Sanderson and C.J. van Rijsbergen. The impact on retrieval effectiveness of skewed frequency distributions. *ACM Transactions on Information Systems*, 17(4):440–465, 1999.

[103] S. Scott and Matwin S. Feature engineering for text classification. In *Proc. of the 16th International Conference on Machine Learning*, pages 379–388, Bled, SLovenia, 1999.

[104] Y. Shinyama and S. Sekine. Paraphrase acquisition for information extraction. In *Proc. of ACL 2003, 2nd Workshop on Paraphrasing: Paraphrase Acquisition and Applications*, pages 65–71, 2003.

[105] R. Sinha and R. Mihalcea. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proc. of the IEEE International Conference on Semantic Computing*, 2007.

[106] G. Siolas and F. dAlche Buc. Support vector machines based on semantic kernel for text categorization. In *Proc. of IJCNN 2000*, pages 205–209. IEEE Press, 2000.

[107] A.F. Smeaton, F. Kelledy, and R. O'Donnell. Trec-4 experiments at dublin city university: Thresholding posting lists, query expansion with wordnet and pos tagging of spanish. In *Proc. of the TREC 1995.*, 1995.

[108] B. Snyder and M. Palmer. The english all-words task. In *Proceedings of Senseval-3*, pages 41–43. Barcelona, Spain, 2004.

[109] Y.I. Song, K.S. Han, and H.C. Rim. A term weighting method based on lexical chain for automatic summarization. In *Proc. of the 5th CICLing Conference*, pages 636–639, USA, 2004.

[110] K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[111] C. Stokoe. Differentiating homonymy and polysemy in information retrieval. In *Proc. of the HLT/EMNLP*, pages 403–410, 2005.

[112] C. Stokoe, M.P. Oakes, and J. Tait. Word sense disambiguation in information retrieval revisited. In *Proc. of the 26th SIGIR*, pages 159–166. ACM, 2003.

[113] M. Strube and S.P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *Proc. of the 21st AAAI*, 2006.

[114] M. Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proc. of CIKM 1993*, 1993.

[115] M. Theobald, R. Schenkel, and G. Weikum. Exploiting structure, annotation, and ontological knowledge for automatic classification of xml data. In *Proc. of the WebDB 2003*, pages 1–6, 2003.

[116] K. Toutanova, D. Klein, C. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of HLT-NAACL*, pages 252–259, Canada, 2003. ACM.

[117] G. Tsatsaronis and V. Panagiotopoulou. A generalized vector space model for text retrieval based on semantic relatedness. In *Proc. of the EACL 2009 (Student Research Workshop)*, pages 70–78, 2009.

[118] G. Tsatsaronis, R. Pitkanen, and M. Vazirgiannis. Clustering for ontology evolution. In *Proc. of the 29th Annual Conference of the German Classification Society (GfKl)*, 2005.

[119] G. Tsatsaronis, I. Varlamis, and M. Vazirgiannis. Word sense disambiguation with semantic networks. In *Proc. of the 11th International Conference on Text, Speech and Dialogue (TSD 2008)*, pages 219–226, September 2008.

[120] G. Tsatsaronis, I. Varlamis, M. Vazirgiannis, and K. Nørvåg. Omiotis: A thesaurus-based measure of text relatedness. In *Proc. of PKDD 2009, Demo Paper (to appear)*, 2009.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[121] G. Tsatsaronis, M. Vazirgiannis, and I. Androutsopoulos. Word sense disambiguation with spreading activation networks generated from thesauri. In *Proc. of the 20th IJCAI*, pages 1725–1730, 2007.

[122] M. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proc. of the 12th European Conference on Machine Learning*, 2001.

[123] P.D. Turney. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416, 2006.

[124] P.D. Turney. The latent relation mapping engine: Algorithm and experiments. *JAIR*, 33:615–655, 2008.

[125] P.D. Turney. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proc. of the COLING 2008*, pages 905–912, 2008.

[126] P.D. Turney and M.L. Littman. Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1-3):251–278, 2005.

[127] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

[128] C.J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.

[129] I. Varlamis, M. Vazirgiannis, M. Halkidi, and B. Nguyen. Thesus: Effective thematic selection and organization of web document collections based on link semantics. *IEEE TKDE Journal*, 16(6):585–600, 2004.

[130] T. Veale. Wordnet sits the sat: A knowledge-based approach to lexical analogy. In *Proc. of ECAI 2004*, pages 606–612, 2004.

[131] J. Veronis and N.M. Ide. Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proc. of the 13th International Conference on Computational Linguistics*, pages 389–394. Finland, 1990.

[132] S.N.V. Vishwanathan and A.J. Smola. *Fast Kernels for String and Tree Matching*. MIT Press, 2003.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

[133] E. Voorhees. Using wordnet to disambiguate word sense for text retrieval. In *Proc. of the 16th SIGIR*, pages 171–180. ACM, 1993.

[134] S.K.M. Wong, W. Ziarko, V.V. Raghavan, and P.C.N. Wong. Modeling of information retrieval concepts in vector spaces. *ACM Transactions on Database Systems*, 12(2):299–321, 1987.

[135] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *Proc. of the 32nd ACL*, pages 133–138, 1994.

[136] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proc. of the SIGIR 1999.*, pages 42–49, 1999.

[137] D. Yarowsky. Unsupervised word sense disambiguation rivalling supervised methods. In *Proc. of the 33rd ACL*, pages 189–196, 1995.

[138] Y. Zhao and Karypis. G. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168, 2005.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ