WORD SENSE DISAMBIGUATION AND CONTEXT

Anahit Martirosyan

Thesis

submitted to the Faculty of Graduate and Postdoctoral
Studies
in partial fulfillment of the requirements
for the degree of Master of Computer Science

May, 2005

Ottawa-Carleton Institute for Computer Science
School of Computer Science
Carleton University

Ottawa, Ontario, Canada

ABSTRACT

The main source of information, contributing to Word Sense Disambiguation (WSD), is the context around a given word. Deciding on the number of context words to take into account for WSD and their distance from the word to be disambiguated are important considerations in the design of WSD algorithms.

This thesis is an experimental study of sense ambiguity and context. The study was based on modifications to the NRC WSD system. We present results of our modified system's performance with different windows of content words around the ambiguous word.

Semantic similarity of two words is a quantitative measure of the degree to which two word senses are related. Our experiments used corpus-based and lexicon-based measures of semantic similarity. In both cases, our results suggest that the number of content words around the ambiguous word must be relatively small to be useful in WSD systems of the kinds that we investigated.

TABLES

FIGURES

APPENDICES

ACKNOWLEDGEMENTS

I would like to acknowledge the help that I have received while working on my thesis.

I am very grateful to my supervisors:

- Dr. Peter Turney, for choosing the topic of my thesis and guiding me at every stage of my work.

- Dr. Jean-Pierre Corriveau, for directing my research and helping me gain self-confidence.

I would also like to thank the members of my committee:

- Dr. D. Inkpen, SITE, University of Ottawa
- Dr. J. Oommen, School of Computer Science, Carleton University
- Dr. M. Smid, School of Computer Science, Carleton University

1 Introduction

This chapter describes the problem of Word Sense Disambiguation (WSD). It then introduces methods for WSD, briefly describes Senseval-2 and its English Lexical Sample task, then reviews the National Research Council (NRC) WSD system, presents the hypotheses of this thesis, reviews the experiments and their results and describes the organization of the thesis.

1.1 Word Sense Disambiguation (WSD)

Identifying the intended sense of an ambiguous word is a hard problem for Natural Language Processing (NLP). Ambiguous words can be homonyms (single spoken and written form that refers to multiple unrelated concepts) or polysemous words (words that have multiple related senses).

For example, word *palm* is a homonym: in the following sentences, it has two different meanings. In the first case, *palm* refers to a *tree* and in the second case, it refers to a part of the *hand:*

Palms grow in a tropical climate.

The baby's *palms* were tiny.

An example of a polysemous word is the word *hard.* In the following two cases, the word *hard* has two distinct (though related) senses:

It is *hard* to keep a secret.

The cookies were *hard.*

In the first case, the word *hard* means *difficult* and in the second case, it means *dried out.*

WSD usually involves the association of a given ambiguous word in the text in which it appears with one of its meanings (senses), based on the surrounding context (words in the text preceding and following the word to be disambiguated) in the text. There is often disagreement about how many senses an ambiguous word has and what they are. WSD researchers avoid this disagreement by selecting a specific computational lexicon (such as WordNet (Miller, 1995)) as their standard for determining the senses of a word. After the lexicon is chosen, the task of WSD consists of determining means to associate each occurrence of a word to its appropriate sense.

1.2 Approaches to WSD

For Symbolic and Connectionist approaches to WSD, determination of word senses relies on hand-encoding of all the possible meanings of an ambiguous word into a knowledge base. Instead of a knowledge base, statistical approaches for WSD use information from electronic dictionaries, thesauri, bilingual dictionaries and sense-labeled training corpora.

Statistical methods of WSD are classified as knowledge-based and corpus-based. WSD consists of comparing the context of the instance of the word to be disambiguated with either information from an external knowledge source (knowledge-based WSD), or information about the contexts of previously disambiguated instances of the word (hand-labeled training data) derived from corpora (corpus-based WSD).

For instance, in the dictionary-based (one of the knowledge-based methods) method of WSD, disambiguation is accomplished by finding the sense of the ambiguous word, for which the words in the dictionary definition have the maximum overlap with the words in the dictionary definitions of the surrounding words.

Most of Corpus-based approaches to WSD use supervised Machine Learning. In supervised disambiguation, hand-labeled training data is available, where each occurrence of an ambiguous word is annotated with a semantic label taking into account the context of the word. Each sense-labeled occurrence of a particular ambiguous word is

transformed into a feature vector. The features are usually derived from the contextual words in a sample of sentences that contain the word to be disambiguated (e.g., the features could include syntactic dependences, co-occurring words). The task of supervised Machine Learning is to build a classifier, which assigns an appropriate sense to the word to be disambiguated in the test data based on its surrounding context. The approaches to WSD are discussed in Chapter 2 in more detail.

1.3 Senseval-2

Senseval is an evaluation exercise for algorithms performing WSD. Senseval is organized by ACL SIGLEX (the Special Interest Group on the LEXicon of the Association for Computational Linguistics) and EURALEX (European Association for Lexicography). There have been three Senseval competitions: Senseval-1 (1998), Senseval-2 (2001), and Senseval-3 (2004). Senseval-4 is currently being planned. Senseval events greatly contribute to active research in WSD field. Senseval also aims at finding which WSD programs perform best, and which words or languages present particular problems to which algorithms of WSD.

The tasks of Senseval-2 (2001) were 'all words', 'lexical sample' and translation tasks. In the training data of the 'all words' task, all the content[1] words in a corpus are sense-labeled. The task is to disambiguate all words in test data. The 'lexical sample' task requires disambiguation of a lexical sample, consisting of 50 to 100 dictionary head words. Training data for the 'lexical sample' task contains texts where a lexical sample is manually sense-labeled and test data contains texts where the WSD algorithm must guess the sense labels for the specified head words. The task in this case is to disambiguate only the lexical samples, not all words in the test data. In the translation task, senses correspond to distinct translations of a word into another language.

[1] Words, such as a noun, verb, or adjective, which have a statable lexical meaning.

The experiments for this thesis were based on modifications to the National Research Council WSD system designed for the English Lexical Sample task of Senseval-2. The task is described in the next section in brief and in Chapter 3, in detail.

1.3.1 English Lexical Sample (ELS) Task of Senseval-2

The requirements of the ELS task of Senseval-2 were to disambiguate 73 words: 15 adjectives, 29 nouns and 29 verbs. Every word of the ELS task was presented with training and test data. The training and test data consisted of examples (paragraphs of text) for each word. The word to be disambiguated was marked in the examples as a *head* word. The head words in training examples were manually sense-labeled by lexicographers with WordNet's senses, according to the surrounding context. The test data was annotated by lexicographers as well, but the answers weren't released to the participants until Senseval-2 was over. The ELS task was to use the sense-tagged training data to assign senses to the head words in the test data.

1.4 Brief Overview of the NRC WSD system

The National Research Council (NRC) WSD system participated in the English Lexical Sample (ELS) task of Senseval-3[2] (the task's objective is similar to the one described in the previous section; it is described in detail in Chapter 3). The system accomplishes WSD by means of supervised Machine Learning. The system represents the words to be disambiguated (head words) as feature vectors with several hundred features, half of which are syntactic, half semantic. Syntactic features are generated from the stop words (a word that has low information content, such as a pronoun, preposition). They describe the position of the stop word with respect to the head word in a window (that is, a certain number of words from the context centered on the target word) of five words centered on

[2] http://www.senseval.org/senseval3.

the head word. Semantic features take into account the semantic content of the head word; they are generated from the content words in a 9-word window centered on the head word. Every head word is represented by two semantic features: one preceding and one following content word. The syntactic and semantic features together form the feature vector of a given head word.

An innovative aspect of the system is the method for generating the semantic features, based on word co-occurrence probabilities using Pointwise Mutual Information – Information Retrieval (PMI-IR) (Turney, 2001). The probabilities of co-occurrence are estimated using the Waterloo MultiText System (Clarke et al., 1995; Clarke and Cormack, 2000; Terra and Clarke, 2003) with a corpus of about one terabyte of unlabeled text. The Weka (Witten and Frank, 1999) Machine Learning software and a rule-based part-of-speech tagger[3] (Brill, 1994) are used in the system.

After each head word in the training and test data was represented by a feature vector, the NRC WSD system used Weka Machine Learning software to learn a model of the training data and make predictions about the classes (senses) of a word to be disambiguated in the test examples. The system is described in detail in Chapter 3.

The NRC-Fine system was one of the four versions of the NRC WSD system evaluated on the ELS task of Senseval-3 (the system and its different versions are described in Chapter 3). The NRC-Fine version of the system will be referred to as NRC WSD system hereafter.

The experiments in this thesis are based on the Senseval-2 ELS data instead of the Senseval-3 ELS data, because the official sense labels for the Senseval-3 test data were not publicly available when we began our experiments.

[3] The tagger assigns part of speech tags to the words (e.g., NN is a part of speech tag for noun).

1.5 Goal and Hypotheses of the Thesis

From the first attempts at automated WSD, which were made in Machine Translation in 1949, the importance of the context, in which an ambiguous word is encountered, was apparent. Weaver (1949) formulated the problem of sense ambiguity and context:

> *If one examines the words in a book, one at a time as through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of the words. [...] But if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say N words on either side, then if N is large enough one can unambiguously decide the meaning of the central word. [...]The practical question is: "What minimum value of N will, at least in a tolerable fraction of cases, lead to the correct choice of meaning for the central word?"*

The goal of this thesis is to modify the NRC WSD system in order to discover the optimal number and position of context words that lead to the best 'performance' of this system. The goal is obtained by means of an experimental study of word sense ambiguity and context. This study centers on three hypotheses, the two first ones addressing 'performance'.

The first hypothesis of this thesis is:

"The accuracy of the modified version of the NRC WSD system improves as more *semantic* features are introduced, by expanding the window of *content* words around the head word."

The hypothesis was tested by creating several modified versions of the NRC WSD system. Each version used a different size of window of content words (starting from 4- to 1-word window size) around a word to be disambiguated. It is necessary to define the term 'window of words' used in our experiments. Usually 'window of *N* words' implies

N words centered on the head word. For our experiments, we consider a window of words, which is asymmetrical and which contains only the content words (i.e., non-stop words; nouns, verbs, and adjectives).

The number of words to the left of the word to be disambiguated varied from three to one, whereas the number of words to the right of the word to be disambiguated was one or zero. The reason for choosing an asymmetrical window stems from the format of ELS data: the head words were almost in all cases located at the end of the last sentence of the paragraph. The ELS data is formatted this way most likely because of the common assumption that humans disambiguate words reading text from left to right. A reader can usually disambiguate a word immediately, without reading ahead, using only what is already read (the left context). Only rarely does a reader need to read ahead (the right context) to disambiguate a word. If there were no content words on either side of the head word, then special null characters were added in place of a word. The idea of our hypothesis is: the more semantic features (via content words)[4] around the head word are taken into account, the better are the chances for disambiguation.

There was no significant difference in the system's accuracy in the experiments with two semantic features (one preceding the head word content word and one following) versus experiments with a larger window size of three or four semantic features (two or three content words before the ambiguous word and one after, respectively). However, using four and three content words, results take longer time to calculate compared with two. Thus, from the point of view of the speed (computation time), the system's performance was better in the case of two semantic features than for three or four semantic features. From these results, we hypothesized that additional semantic features (e.g., the two furthest words from the head word in a four-word window) could be less relevant to the process of disambiguation of the head word than the two nearest context words. An alternative hypothesis was that the additional features could be relevant but redundant (i.e., they do not add more information for disambiguating a head word compared to the two nearest words). In other words, the relevance of the context words decreases as their

[4] For simplicity, hereafter we will use the expressions 'semantic feature' and 'content word' interchangeably.

distance from the head word increases. Such considerations led to our second hypothesis:

"Taking into account additional semantic features in the NRC WSD system, by expanding the window of content words around the head word, does not improve accuracy because the new features are irrelevant for disambiguating the head word."

The last hypothesis of the thesis claims that the system's performance is not dependent on the method used for calculating semantic feature values. That is, we hypothesize that the relation between context window size and WSD accuracy (observed in our experiments while testing the first two hypotheses) is not an artifact of the way the semantic features are calculated. For testing the third hypothesis, we used WordNet's measures of semantic relatedness as an alternative method for calculating semantic feature values. Thus, experiments of the thesis consisted of two parts: in the first part, we used a corpus-based measure of semantic similarity (PMI-IR) for calculating semantic feature values, and in the second part, lexicon-based measures of semantic similarity (WordNet measures of semantic relatedness), for the same purpose. The third hypothesis of this thesis is:

"If the NRC WSD system is modified by replacing the statistical measure of semantic similarity (PMI-IR) with lexical measures of similarity (using WordNet), Hypothesis 1 will still be rejected and Hypothesis 2 will still be accepted."

The results of our experiments are described briefly in the next section and in detail in Chapter 5.

1.6 Overview of Experiments and Results

Experiments were based on the ELS task from Senseval-2. A subset of the Senseval-2 English Lexical Sample (30 words: 10 nouns, 10 adjectives and 10 verbs) was used for all experiments with our modified versions of the NRC WSD system, with the purpose of reducing the computation time required for running the experiments with the full set of words. The experiments used a different number of semantic features from a four-word

window (as mentioned in the previous section) around the head word. Modified versions of the NRC WSD system considered the following number and positions of semantic features:

- four semantic features (three content words before the head word and one after)
- three semantic features (two content words before the head word and one after)
- two nearest semantic features (one content word immediately before the head word and one immediately after)
- two furthest semantic features (two furthest content words from 4- word window)
- one semantic feature (one content word before the head word)
- zero semantic features, only syntactic features.

In the experiments for testing the *first* and *second* hypotheses, the modified versions of the NRC WSD system were evaluated using a statistical measure of semantic similarity (PMI-IR) for calculating the semantic feature values. Statistical significance tests on the results of the experiments for testing the *first* hypothesis showed that the best results (taking into account the accuracy of the system and its computation speed) are obtained when the system uses the two nearest semantic features. When three and four semantic features are taken into account, the system's accuracy does not improve and computation time increases. When the system takes into account one semantic feature or no semantic features, then there is a significant drop in the system's accuracy. Thus, the first hypothesis of the thesis, namely that the system has better performance when more content words from the context of a word to be disambiguated are taken into account for our experiments, is not supported.

In the experiments for testing the *second* hypothesis, statistical significance tests on the results with the two furthest and two nearest semantic features demonstrated that the two furthest content words were less relevant to disambiguating the head word compared with the two nearest words. Thus, the second hypothesis of the thesis, about the additional words being less relevant compared to the two nearest words, is supported.

In the experiments for testing the *third* hypothesis, lexicon-based (WordNet) measures of

semantic relatedness were used for calculating semantic feature values. The results of the experiments are similar to the ones using PMI-IR, confirming that the highest accuracy and computation speed are occurring with a two-word window size. Thus, Hypothesis 1 is rejected and Hypothesis 2 is accepted for our WordNet experiments. Consequently, the third hypothesis, about the system's performance being independent of the measure of semantic similarity used in the experiments, is supported.

The detailed results of the experiments are presented in Chapter 5. Observations drawn from the results and comparison of performance of the modified versions of the NRC WSD system using PMI-IR and WordNet similarity measures are discussed in that chapter as well.

1.7 Organization of the Thesis

The thesis is organized as follows:

Chapter 2 describes different concepts of context, and reviews statistical approaches to WSD.

Chapter 3 first reviews the WSD evaluation exercises Senseval-2 and Senseval-3 and their ELS tasks; it then describes the NRC WSD system. The Weka Machine Learning software that is used in the system is described next. The NRC WSD system uses a combination of five basic algorithms from Weka. Two of the five (ADTree and JRip) are presented. The semantic similarity measure PMI-IR and WordNet-based measures of semantic relatedness are described last.

Chapter 4 presents the problem of context. It discusses different concepts of context discussed in the literature on WSD and reviews a few studies on ambiguity and context. The hypotheses of the thesis are then presented, followed by the methodology for testing each hypothesis.

Chapter 5 presents the experiments for testing each hypothesis of the thesis. The results of the experiments for testing Hypothesis 1, Hypothesis 2 and Hypothesis 3 and statistical significance tests' results follow. The chapter concludes with a discussion of the results and a comparison with related studies.

Chapter 6 presents a summary of the thesis, presents conclusions and observations from the studies on ambiguity and context discussed in the thesis, and presents some ideas for future work.

2 Approaches to Word Sense Disambiguation

This chapter first introduces different concepts of context. It then reviews statistical approaches to WSD. The approaches are presented in two sections, which describe corpus-based and knowledge-based methods for WSD.

2.1 What is Context?

The contextual information is the most important source of information for disambiguation. There are two groups of approaches to using the context of an ambiguous word: *bag of words* approaches and *relational* approaches.

In the *bag of words* approaches, context is considered to be the words around the word to be disambiguated in some window, without taking into account their relationships to the target word (e.g. distance, grammatical relations).

In the *relational* approaches, the context around the target word is considered, taking into account the relational information of the words with respect to the target word. The information can include the distance (number of words away from the target word) from the target word, syntactic relations and collocations (collocation is "the occurrence of two words in some defined relation" (Yarowsky, 1993)).

The following sentences are examples of 'verb-subject' syntactic relations for the verb *keep*:

He *kept* eating.

He *kept* calm.

He *kept* a record.

In the first sentence, the subject of the verb is a gerund, in the second and third cases the

subject of the verb is an adjectival phrase and a noun phrase respectively.

An example of collocation is the phrase 'football match'. Here the word *football* occurring with the word *match* implies that the meaning of match in this phrase is *game*.

Different categories of context are defined in the literature on WSD. They mainly differ on the number of words taken into account during disambiguation and their distance from the target word. Three main categories of context are: local context, topical context and domain information (Ide and Veronis, 1998). Local context generally takes into account a small number of words surrounding the target word, limited to the sentence where the target word is encountered. Topical context includes words from a few sentences around the target word. Domain information is used for deciding to which domain the target word belongs and it includes words from the whole document in which the target word occurs. The problem of context and some studies of WSD based on different categories of context are presented in Chapter 4.

The next sections review statistical methods of WSD. Statistical methods now predominate over the older Symbolic and Connectionist approaches to WSD, hence the older approaches are not discussed in this thesis. An extensive survey on Symbolic and Connectionist approaches to WSD is presented in Ide and Veronis (1998).

2.2 Statistical Methods of WSD

Statistical approaches to WSD use large-scale lexical resources such as dictionaries, thesauri, and corpora. They are classified as corpus-based (section 2.2.1) and knowledge-based (section 2.2.2).

2.2.1 Corpus-based WSD

This section presents corpus-based methods for WSD. Supervised WSD is presented first

(section 2.2.1.1), followed by Unsupervised WSD (section 2.2.1.2).

2.2.1.1 Supervised WSD

In Supervised disambiguation, a manually-sense-tagged corpus is available for training. A corpus is a collection of samples of previously disambiguated texts. Machine Learning algorithms are applied to learn statistical models from corpora in order to perform WSD. Thus, disambiguation rules are learned from a manually-sense-tagged corpus. During training on a disambiguated corpus, probabilistic information about the context words (frequency of occurrence of context words in a particular sense of the ambiguous word) as well as distributional information about the different senses (frequency of a word used in the corpus in each of its senses) of an ambiguous word is collected. Distributional information and context words are used to define senses of ambiguous words. In the test phase, the sense with highest computed probability based on the training data is chosen. Sense-tagging of a corpus is costly and time-consuming as it is done manually.

2.2.1.1.1 Machine Learning for WSD

Machine Learning can be interpreted as "the acquisition of structural descriptions from examples" (Witten, Frank, 2000). A critical component of any application of Machine Learning is the representation of the training examples and the generated model: the classifier (the 'disambiguator', in the case of WSD). Figure 1 illustrates the process of training. The classifier is built from the training examples.

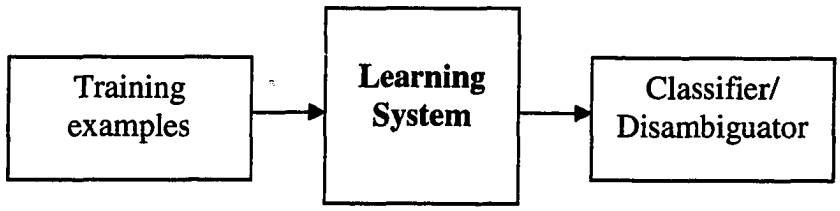

Figure 1 Learning System.

Supervised disambiguation can be viewed as an instance of statistical classification. The task is to build a classifier, which classifies new cases (new occurrences of the ambiguous word) based on their context in the corpus. Figure 2 illustrates the process of testing, when examples from the test data are presented to the classifier and the classifier assigns word senses to the words to be disambiguated.

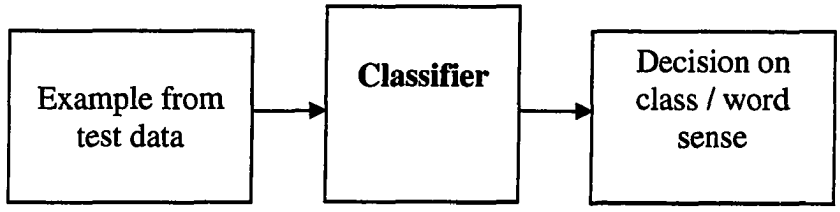

Figure 2 Classification system.

A commonly used representation for training examples in Machine Learning is a feature vector, that is, a "fixed set of features, taking values from a fixed set" (Paliouras et al., 2000). Deciding on what features will be used, is crucial for Machine Learning. Also, the question of how much context should be taken into account to construct the classifier plays an important role in WSD. Many systems interpret context differently. (Different concepts of context are discussed in Chapter 4.) Examples of features used in WSD are collocated words within a window of words surrounding the ambiguous word, and syntactic features, describing the examples of training data. Examples of collocation features of a given word include noun phrases (e.g. "white wine") and phrasal verbs (e.g. "to make up"). Syntactic features can describe the grammatical structure of the sentence in which the ambiguous word is encountered (such as object-verb, verb-subject

relationships etc.). These types of features dominate the literature on learning methods for WSD.

Two Machine Learning algorithms implemented in the NRC WSD system are described in the next chapter.

2.2.1.1.2 Disambiguation based on Translation in a Second Language Corpus

WSD based on translation makes use of word correspondences in a bilingual dictionary. The language of the text that is to be disambiguated is referred to as a first language and the target language in the bilingual dictionary as a second language. First language text will be disambiguated using second language corpus.

The algorithm of Dagan and Itai (1994) first identifies the translations of the senses of an ambiguous word into a second language. The phrase where the ambiguous word is used is translated into a second language as well. The second language corpus is searched for the translation of the phrase. If the phrase occurs with only one of the translations of the word, then the ambiguous word is assigned the corresponding sense. An example in Figure 3 from Manning et al. (1999) demonstrates the algorithm used for disambiguation of English word *interest* using a German corpus:

	Sense 1	Sense 2
Definition	legal share	attention, concern
Translation	*Beteiligung*	*Interesse*
English collocation	acquire an interest	show interest
Translation	*Beteiligung erwerben*	*Interesse zeigen*

Figure 3 Example of WSD by using a second language corpus.

If *interest* is used in the phrase *showed interest,* German translation of *show,* 'zeigen' will occur only with *Interesse.* The conclusion is that *interest* in the phrase to *show interest*

belongs to the sense *attention, concern.* And translation of the phrase *acquired an interest* is *erwarb eine Beteiligung*. Thus, *interest* will be disambiguated by being assigned its second sense, *"legal share"*.

This method has some limitations since translations of many ambiguous words into a second language can in turn be ambiguous themselves. Another difficulty is the availability of second language corpora.

2.2.1.2 Unsupervised WSD

Completely Unsupervised WSD is used when no supporting tools such as dictionary, thesaurus, and sense-tagged corpus are available; for example, when dealing with information from specialized domains, for which there may be no lexical resources available. Completely unsupervised WSD is not possible in terms of sense tagging; occurrences of a word must be tagged with one of the possible senses. Sense discrimination is possible in an unsupervised manner; it is possible to cluster the contexts of an ambiguous word then discriminate between the groups. One of the algorithms for unsupervised WSD (Schuetze, 1998) consists in context-group discrimination: to cluster contexts of an ambiguous word into a number of groups and discriminate between them without labeling them.

2.2.2 Knowledge-based WSD

WSD in these approaches uses lexical resources such as machine-readable dictionaries, thesauri and bilingual dictionaries.

2.2.2.1 Dictionary-based Disambiguation

Lesk (1986) uses machine-readable dictionaries to obtain the definitions of a word and to use them for WSD. Every word is represented by its *signature,* which consists of a list of the words occurring in the definition of the word. Disambiguation is accomplished by means of finding overlaps in the signature of an ambiguous word with signatures of its neighbours in the context.

For example, a word *cone* has two definitions in a dictionary:

1. a fruit of pine family tree;
2. a geometric shape.

If either *forest* or *math* occurs in the same context with *cone,* then it will be disambiguated by looking for overlaps in the definitions of *cone/forest* or *cone/math* respectively. In the first case, the first sense of *cone* will be inferred and in the second case, its second sense.

Dictionary-based methods, even though drawing on a large-scale information source, have some shortcomings such as poor coverage of words in certain domains, and the fact that dictionaries are created for human use and not for machine exploitation. Above-mentioned methods are very sensitive to the exact matching of lists of words in the definitions. If some of the words in the definitions are missing, the results of assigning a word to an appropriate sense might change.

Currently the mostly used electronic dictionary is WordNet. Chapter 3 talks about WordNet and WordNet-based similarity measures. WordNet similarity measures are used for WSD as they assign a quantitative measure of the degree to which two word senses are related for a pair of words.

2.2.2.2 Thesaurus-based Disambiguation

Thesaurus-based disambiguation uses semantic categorization of a thesaurus like Roget's. Thesauri contain information about relationships among words such as synonymy.

Each occurrence of the same word under different categories of the thesaurus corresponds to different senses of that word. The words in the same category of thesaurus are semantically related and are organized into level hierarchy.

The basic idea, which is implemented in the thesaurus-based methods for WSD (Yarowsky, 1992), is that the semantic categories of the words in context determine the semantic category of the context in whole and that this category in turn determines which word sense of an ambiguous word is used.

Each word is assigned one or more subject codes in the thesaurus. If a word is ambiguous then it is assigned several subject codes. They correspond to different senses of a word.

An ambiguous word can be disambiguated by counting the number of words in the context with the same subject code in the thesaurus. This subject code, i.e., the category, will be considered a possible topic of the context. The category, which scores the maximum number of counts, wins. Therefore, an ambiguous word will be assigned a sense from that category. For example, the word *bass* has two senses and belongs to two categories in Roget's thesaurus such as:

Sense	Roget category
musical senses	Music
fish	Animal, insect

Depending on which category the majority of the words in context belongs, either sense of the word *bass* will be chosen. In the next two sentences, the word *bass* will be assigned to different senses by examining to which categories the surrounding words in context belong:

An electric guitar and *bass* players of the band were on the stage.

Their fishing rod was good and they caught a salmon and a *bass*.

In the first sentence, the words such as *guitar, player, band* belonging to the category Music, will contribute to assigning the first sense to the word *bass*.

In the second sentence, the words such as *fishing, rod, salmon* belonging to category Animal, will result in assigning the second sense to the word *bass*.

General topic categorization may be problematic because of coverage: new concepts are not to be found in older thesauri. An algorithm for adaptation of a topic classification to a corpus (Yarowsky, 1992) may be used to overcome this problem. For each word of a category in the thesaurus, a 100-word context is extracted from the corpus. By using statistics, words that are most likely to occur with words from the category are identified from corpora. The words are then added to the category of the thesaurus.

Thesauri provide valuable information about word categorization and word relations, which is used to resolve word sense ambiguity. Like machine-readable dictionaries, thesauri are created for humans and therefore are not a best source of information for automatic WSD. In particular, higher levels of a concept hierarchy are sometimes so broad that they are not useful in creating semantic categories.

2.2.2.3 One Sense per Discourse, One Sense per Collocation

The dictionary-based methods process each occurrence of an ambiguous word separately. In contrast, Yarowsky (1995) emphasizes some common constraints between different occurrences of an ambiguous word. More specifically, his approach focuses on two observations. The first observation, one sense per discourse, is that the sense of a target word is highly consistent within a document (an ambiguous word occurring several times within a document is likely to have the same sense). The second observation, one sense per collocation, is that words surrounding the target word provide strong and useful

information for the word's disambiguation. This information depends on a relative distance to the word, order and syntactic relationship (examples of collocations were presented above).

Yarowsky's method uses the above-mentioned properties (one sense per discourse, one sense per collocation) to incrementally identify collocations for target senses of a word, given a few seed collocations (examples of collocations) for each sense, either from dictionary definitions or a small hand-labeled training set.

A context of a target ambiguous word is taken and compared with a given set of collocations for each sense of an ambiguous word. If the same collocations are found in the context of the target word, then this collocation becomes a candidate for disambiguation of the target word. The process iterates with all the contexts. The strongest collocation is then chosen for each context. After this part of the algorithm, the constraint 'one sense per discourse' is applied. All instances of the ambiguous word in the text are assigned to the majority sense in a document.

This algorithm works when material to be disambiguated is a collection of small documents. The advantage of this algorithm is that it does not need a labeled set of training examples, but just a few collocation seeds.

3 The NRC WSD System

This chapter first describes English Lexical Sample (ELS) task of Senseval-2 and Senseval-3. It then reviews the NRC WSD system used for the Senseval-2 ELS task, followed by the system's description for the Senseval-3 ELS task. The Machine Learning tool WEKA used by the NRC WSD system is briefly introduced next. Two of the Machine Learning algorithms implemented by the system, namely decision tree induction and decision rule induction, are then described. Semantic similarity measure PMI-IR and WordNet-based measures of semantic relatedness conclude the chapter.

3.1 English Lexical Sample (ELS) Task of Senseval-2

As mentioned in Chapter 1, the requirements of ELS task of Senseval-2 were to disambiguate 73 words: 15 adjectives, 29 nouns and 29 verbs. Every word to be disambiguated was presented with cases of training data and test data. Each case consisted of a paragraph of text, where the words to be disambiguated were marked as *head* words. Those words were manually sense-labeled with WordNet's senses appropriate to the context of the word in the training data. The head words in the test data were sense-labeled, but the labels were hidden from the WSD systems. The hidden senses were revealed after every team had submitted their 'solutions' for the test data. The hidden manually assigned sense labels in the test data were used to score the performance of the WSD systems. The task of ELS was to use the sense-labeled training data to assign senses to ambiguous words in the test data.

Figure 4 presents examples of sentences from the training cases, which consisted of a paragraph of text. In the majority of cases, the head word was located in the last sentence of the paragraph:

-The management of information becomes an *<head>art</head>*,
master it and it becomes your unequal advantage.[sense: skill]
-She always had an interest in the *<head>arts</head>*, but in painting

she is self-taught. [sense: arts]

Figure 4 Sentences containing the head words from training data of Senseval-2's ELS task.

Some words had both simple and phrasal senses. For example, the word *art* had simple senses such as 'skill' and 'arts' and phrasal senses such as 'art gallery'. The phrase *art gallery* is a collocation: the word *gallery* occurring with the word *art* has only one sense 'art gallery'.

After the participants in the ELS task ran their programs on the test data using the sense-labeled training data, the results were scored by the organizers of Senseval-2. Systems were scored for precision and recall. Precision is defined as a ratio of the number of correctly disambiguated words to the number of words the system attempted to disambiguate (as systems sometimes skipped some words). Recall is defined as a ratio of the number of correctly disambiguated words to the number of words required by the task in total.

Precision and recall are evaluated using different schemes: fine-grained and coarse-grained. For fine-grained values of precision and recall, the answers presented by the system should exactly match the senses, which had been manually assigned by the lexicographers to the head words in the test data. For coarse-grained score, the answers presented by the system are mapped into coarse-grained senses and are compared to the coarse-grained senses previously assigned to the head words in the test data.

Coarse-grained senses are clusters of fine-grained senses. They were created by the organizers of Senseval. For example, WordNet lists four senses for the word "art". Thus, "art" has four fine-grained senses in WordNet and in Senseval-2:

1. the products of human creativity, works of art;

2. the creation of beautiful or significant things;

3. a superior skill that you can learn by study and practice and observation;

4. photographs or other visual representations in a printed publication.

These four senses were *manually* clustered into two clusters (one cluster consists of the senses 1 and 4 and the second cluster consists of senses 2 and 3). These two clusters are the two coarse-grained senses of "art".

3.2 ELS Task of Senseval-3

For Senseval-3 ELS task, the nouns and adjectives were sense-tagged using WordNet senses, the same as they were in Senseval-2. Verbs were annotated with the senses from Wordsmyth[5]. A different source of sense inventory for verbs was used because of poor performance of verbs in the systems participating in the ELS task of Senseval-2. The main change in Senseval-3 consisted of a new way of collecting annotated data. The data was sense-labeled using contributions from Web users as opposed to the previous Senseval tasks, where the data was sense-labeled by lexicographers. The purpose of the new method was to overcome lack of annotated data and difficulties related with lexicographers' work. The requirements of the ELS task of Senseval-3 were the disambiguation of 57 words (each of the words having around 6 senses) using 140 training examples and 70 test examples. The words to be disambiguated were marked as *head* words. Every head word was presented with a set of examples, where a word was used in different senses. In the training data, the head words were sense-labeled. The head words were also sense-labeled in the test data, but, as usual, these labels were hidden from the WSD algorithms and the teams participating in Senseval-3, until the end of Senseval-3. The sense labels for the test data were used to evaluate the accuracy (precision and recall) of the WSD systems that participated in Senseval-3. Similarly to ELS tasks of previous Senseval competitions, the task of ELS was to assign senses to the head words in the test data.

[5] http://www.wordsmyth.net/

3.3 The NRC WSD System for Senseval-2

As briefly described in Chapter 1, the NRC WSD system (Turney,2004) treats the ELS task as a supervised Machine Learning problem. The system learns a model from the training data and assigns classes (senses) to the head words in the test data. Each training example consisted of about a paragraph of text, with the head word marked and sense-labeled. The examples of the training data, corresponding to one head word, are converted into feature vectors. The vectors consist of syntactic and semantic features. Every head word has its own unique feature vector. The same procedure is done for the test data. After the feature vectors are created, the Weka Machine Learning tool (described in Section 3.5) is used to learn a model of training data and predict the classes of test data.

First, the system uses Brill's rule-based part of speech tagger (Brill, 1994) to assign part-of-speech tags to the words in a training example. A nine-word window from the tagged text is extracted (four words before and four words after the head word). The window is not allowed to go beyond a sentence. If there are not enough words within a sentence, then special null characters are inserted in the positions of the missing words. When each example of training data is converted into a nine-word window, it is then used for generating feature names for the head word. First, the names of syntactic and semantic features are generated from the training data. The feature values for the training and test data are calculated later. The feature names show how the features will be calculated.

After the same process with part-of-speech tagging and extracting the windows is done with the test data, syntactic feature values will be defined for both training and test data. The names of syntactic features have the form: *matchtype_position_model.* A five-word sub-window of a nine-word window is used for generating syntactic feature names. There are five positions: hm2 (head word minus two), hm1 (head word minus one), hd0 (head word), hp1 (head word plus one) and hp2 (head word plus two).

There are three matchtypes: *ptag* (partial tag), *tag* and *word.* The syntactic feature names are generated by all possible combinations of matchtype, position and model. Partial tag

ptag means that similar parts of speech will be counted as the same when the feature values are defined in the training and test data. For example, if a feature is *ptag_hm1_VB* then "VB" (Verb, base form) and "VBD" (Verb, past tense) will be counted as the same[6]. A *tag* matchtype requires stricter match in the matchtype and model: feature name *tag_hm1_VB* will count only part of speech "VB" in the position *hm1*. Feature names for matchtype *word* are generated based on the training example. For matchtype *word,* the *model* is not allowed to be a noun, verb or adjective. Those words are used for generating semantic features.

Syntactic features have binary values: zero or one. The feature of the form *matchtype_position_model* for a given window (of training or test data) will have a value one if there is a match of the type *matchtype* in the position *position* with the *model* word. Otherwise, it will be set to zero. For example, feature *tag_hm1_VB* will have the value one if the window contains a verb *(VB)* in the position preceding the head word by one word *(hm1).*

The semantic feature names are of the form: *position_model.* There are two positions: *pre* (preceding) and *fol* (following). The *model* word is a content word (i.e., noun, verb or adjective) extracted from the training data. It is the content word that is either preceding or following the head word. For example, if the nearest noun, verb or adjective, preceding the head word, is "museum", then the feature name will be *pre_museum.* Another kind of semantic features has the name: *avg_position_sense. Position* is the same as in previous semantic feature names and the *sense* can be any label of the sense in which the head word is used.

The values of semantic features are real numbers. The value of the feature *position_model* is calculated by measuring the semantic similarity between the word in the position *position* in the training or test data and *model* word. The semantic similarity between two words is measured by their Pointwise Mutual Information, $\mathrm{PMI}(w_1, w_2)$, using Information Retrieval:

---

[6] VB and VBD are examples of part of speech tags assigned by Brill's part of speech tagger.

$$\mathrm{PMI}(w_1, w_2) = \log_2(p(w_1 \wedge w_2)/(p(w_1)p(w_2))). \qquad (3.1)$$

A high positive value of $\mathrm{PMI}(w_1, w_2)$ indicates that the words are statistically dependent; they tend to co-occur in a document and thus are likely to be semantically related. When the two words are statistically independent, the probability of their co-occurrence $p(w1 \wedge w2)$ in a document is defined as $p(w1)p(w2)$ and the value of (3.1) is zero. In this case, the words are not semantically relevant. When occurrence of one of the two words implies absence of the other word, then PMI has a negative value. In this case, the words are semantically irrelevant (they do not tend to have any semantic relation). PMI-IR as a measure of semantic similarity is discussed in Section 3.6.1.1 in more detail.

The probabilities in the above equation are calculated, based on the information obtained from querying the Waterloo MultiText System with a corpus of about one terabyte of unlabeled text. The neighborhood size for co-occurrence of words was set to 20 words. It means that the words were searched for co-occurrence within a 20-word window in the same document.

For example, if the word *art* is following the head word in the window, then the semantic feature with the name *fol_exhibition* will have a value of *PMI(art,exhibition),* which indicates semantic similarity between the two words. If there are no nouns, verbs or adjectives in the position *fol* in a window then the value of the feature will be zero.

The values of semantic features of the form *avg_position_sense* are the average values of all the features with the position *position,* for which the model word was extracted from the training examples with the sense label *sense.* For instance, *avg_fol_sense1_01_02* feature will have an average value of all the features *fol_model,* where *model* word was extracted from the training example, which had been labeled with the sense *sense1_01_02.*

Semantic features with the names *position_model* are normalized by being converted to percentiles. Percentile normalization was found useful for another application where features based on PMI-IR were used for supervised learning (Turney, 2003). The

preceding and following features are normalized separately. The *avg_position_sense* features are not normalized as they are calculated after the other features are normalized.

The Weka Machine Learning software was used to induce a model of the training data and classify the cases in the test data. Five classifiers (the –B ones) of Table 1 below were combined by voting; each classifier voted for its best ranked sense, and the sense with the most votes was output as the best guess for the head word in the given test case.

weka.classifiers.meta.Bagging
-W weka.classifiers.meta.MultiClassClassifier
-W weka.classifiers.meta.Vote
-B weka.classifiers.functions.supportVector.SMO
-B weka.classifiers.meta.LogitBoost -W weka.classifiers.trees.DecisionStump
-B weka.classifiers.meta.LogitBoost -W weka.classifiers.functions.SimpleLinearRegression
-B weka.classifiers.trees.adtree.ADTree
-B weka.classifiers.rules.JRip

Table 1 Weka (version 3.4) commands for processing the feature vectors (Copied from Table 1 of Turney, 2004.)

The example Figure 5 illustrates the process of generating semantic features and calculating their values:

Training data				*Test data*		
preceding	<HEAD>	following	sense	preceding	<HEAD>	following
National	<Art>	gallery	fine art	learning	<art>	carpentry
Council	<art>	centre	fine art			
mastering	<art>	canoeing	skill			

Semantic feature names for head word 'art' will be:
<pre_National, pre_Council, pre_mastering, fol_gallery, fol_centre, fol_canoening >.

The feature values for the case from training data *National <Art> gallery:*

<PMI(National, National), PMI(gallery, gallery), PMI(National, Council), PMI(gallery, centre), PMI(National, mastering), PMI(gallery, canoenig) > [fine art].

The feature values for the test case *learning <art> carpentry:*

< PMI(learning, National) , PMI(carpentry, gallery), PMI(learning, Council), PMI(carpentry, centre), PMI(learning, mastering), PMI(carpentry, canoening) >.

Figure 5 Demonstration of the generation of semantic features.

During machine learning, the model learned from values of training feature vector is induced and applied to the test vector of feature values to predict the sense of the head word in the test case.

Phrasal senses (described in Section 3.1) are treated differently from simple senses. For every occurrence of the head word in the test data, the system first checks whether the word is used in a collocation (for example, whether the word *art* is followed by the word *gallery*). The list of all collocations for a head word is created from WordNet (for example, some of the phrasal senses for the word *art* in WordNet are: art gallery, art class, fine art). If a given occurrence of the head word is a collocation then the word is assigned the corresponding phrasal sense (the sense "art gallery", in our example). Otherwise, the sentence containing the head word is processed as described above; by generating a feature vector and using machine learning for guessing the sense of the head word.

All semantic features were evaluated by calculating PMI between the feature and the head word. Those semantic features, which had PMI value lower than a certain threshold value were dropped.

The fine-grained and coarse-grained scores for the NRC WSD system with the data of the ELS task of Senseval-2 are 66.3 and 73.1 respectively.

3.4 The NRC WSD System for Senseval-3

The main change to the NRC WSD system for ELS task of Senseval-3 was due to the fact that the words in ELS task of Senseval-3 had only simple senses; no words with phrasal senses were present in the data. The organizers of Senseval-3 decided to separate the task of identifying the words with phrasal senses (multi-word expressions) from the task of WSD. The words with phrasal senses are usually not ambiguous and their identification is regarded as a separate task from WSD.

Thus, the system processed each head word as described in the previous section by generating feature vectors and using machine learning for assigning senses to head words in test data.

The NRC WSD system had four different versions, participating in the ELS task of Senseval-3: NRC-Fine, NRC-Fine2, NRC-Coarse and NRC-Coarse2. The NRC-Fine2 and NRC-Coarse2 had a different threshold value for dropping features, allowing more semantic features.

The purpose of NRC-Coarse and the NRC-Coarse-2 was to use the coarse-grained senses instead of fine-grained senses during the training. The training examples were relabeled by substituting the fine-grained senses with their corresponding coarse-grained senses.

Table 2 illustrates the results of the NRC WSD system's four entries. The maximum score is obtained with the NRC-Fine version of the system.

System	Fine-Grained Recall	Coarse-Grained Recall
Best Senseval-3 System	72.9%	79.5%
NRC-Fine	69.4%	75.9%
NRC-Fine2	69.1%	75.6%
NRC-Coarse	NA	75.8%
NRC-Coarse2	NA	75.7%
Median Senseval-3 System	65.1%	73.7%
Most Frequent Sense	55.2%	64.5%

Table 2 Comparison of the NRC-Fine with other Senseval-3 ELS systems. (Copied from Table 2 of Turney, 2004.)

From 47 systems, which were presented by 27 teams participating in ELS task of Senseval-3, the NRC WSD system's results were half-way between the best and median.

3.5 Machine Learning Tool WEKA

WEKA (Waikato Environment for Knowledge Analysis) is open source machine learning software[7]. Machine Learning provides the technical basis for data mining. By using data mining "implicit, previously unknown, and potentially useful information from data is extracted" (Witten and Frank, 2000). Regularities and patterns are searched in the data, which are then generalized to make predictions on new data.

One of the goals of learning from previous experience is to extract a decision rule from sample data that could be applied to new data. In supervised machine learning, the learning algorithm requires sample data that has been classified by an expert. This data is called training data. The common objective of building a classifier is to learn from samples and generalize to new cases.

The derived rules, describing the training samples are of the form: *If X is true and Y is false, conclude class 1.* They are evaluated on a set of samples to be either true or false. Each decision rule is associated with a particular class, and a rule which is satisfied (i.e., it is true) means that a particular case corresponds to the particular class.

[7] http://www.cs.waikato.ac.nz/~ml/weka/

A wide variety of supervised machine learning algorithms is implemented in the WEKA software. Two of them, which are used in the NRC WSD system, are ADTree (decision tree induction) and JRip (decision rule induction). The next sections first describe some methods of estimating the error rate of a learning system and then review the two algorithms.

3.5.1 Error Rate Estimation

One of the objectives of learning classifications from sample data is to classify and predict on new data. An error occurs when the classifier assigns a wrong class to a new case. This error rate is called true error rate (or test set error rate). A classifier can also make errors with old cases. The error rate in this case is referred to as an apparent error rate (or training set error rate). An error rate is defined as a ratio of the number of errors to the number of all the cases. For calculating the training set error rate, the number of errors made on the training data and the size of the training data are considered. Conversely, in the case of the test set error rate, the errors made on the test data and the size of the test data are considered:

Error rate = number of errors / number of cases.

The true error rate in general is much higher than the apparent error rate. This happens when the classifier is over-fitted (or over-specified) to the particular characteristics of the training data. Over-fitting happens when a classifier follows the training data very strongly. An extreme case would be when training data itself is used as a classifier; the classification would simplify to simple "table look up" (Weiss, Kulikowski, 1991), to find the correct answer for classification. The apparent error rate will be zero. When new data is presented to such a classifier it will not be able to classify it, as chances of finding an identical case in the table are extremely small because of a huge number of possible combinations of the parameters in the data. When a classifier is over-fitted it means that some features which are irrelevant to making classification rules are present in it. Those irrelevant features (or noise data) will increase the true error rate of the classifier.

The measure of a classifier's success is accuracy. Accuracy of a classifier is computed using the formula: $1-$*Error rate*. Measures of classifier's effectiveness are Precision and Recall (see below). A *contingency table* is used to distinguish different types of errors of the classifier. For example, a classifier is required to make n binary decisions, each of which has exactly one correct answer (either *Yes* or *No*). The result of n such decisions can be summarized in a contingency table, as shown in Table 3. Each entry in the table shows the number of decisions of a specified type. The number of correct decisions for each class falls along the diagonal of the table. For example, a is the number of times the classifier decided *Yes*, and *Yes* was the correct answer; d is the number of times the classifier decided *No* and *No* was the correct answer. By default, it is assumed that there are two classes, *Yes* (positive, relevant) and *No* (negative, irrelevant), and it is assumed that we are only interested in the positive class. The number of times the classifier decided *Yes* is $a+b$; the number of times it decides *No* is $c+d$. The number of times *Yes* was the correct answer is $a+c$; and the number of times *No* was the correct answer is $b+d$.

	Yes is Correct	No is Correct	
Decides Yes	a	b	$a+b$
Decides No	c	d	$c+d$
	$a+c$	$b+d$	$a+b+c+d=n$

Table 3 Contingency table for a set of binary decisions.

Given the contingency table, important measures of the classifier's effectiveness such as Precision and Recall are defined as:

Precision $=a/(a+b)$; Recall $=a/(a+c)$.

Thus, Precision is defined as a ratio of the number of times a classifier assigned a positive class (Yes) correctly to the number of times it assigned that class in total. Recall is defined as a ratio of the number of times the classifier assigned a positive class correctly to the number of times the assignment of the particular class was correct.

Precision and Recall can be defined for each class. There may be cases in which we are interested in the negative class. Then the Precision and Recall are defined as:
Precision = $d/(c+d)$; Recall = $d/(b+d)$.

If there are N classes, where N is greater than 2, then it is possible to calculate Precision and Recall numbers for each of the N classes.

3.5.2 Decision Tree Induction

Decision tree induction is one of the methods used for partitioning a given set of training data (Witten and Frank, 2000). A decision tree partitions data. Figure 6 is an example of a binary decision tree, where each node corresponds to a single test of a condition. In the case of a binary tree, the condition can be either true or false. The starting node is called the root node. If the result of a test in a given node is true or false, the tree will branch either right or left. The final nodes are called the terminal nodes, where a class assignment will be made. For any tree, one path to a terminal node corresponds to a decision rule that is a conjunction (AND) of various tests. For each terminal node there is one path and one decision rule. If there are multiple paths in the tree for the class then the tree is equivalent to the disjunction (OR) of each of the rules (formed from each of the paths). Any decision tree partitions the samples into mutually exclusive paths: for any new case, only one path in the tree will be satisfied.

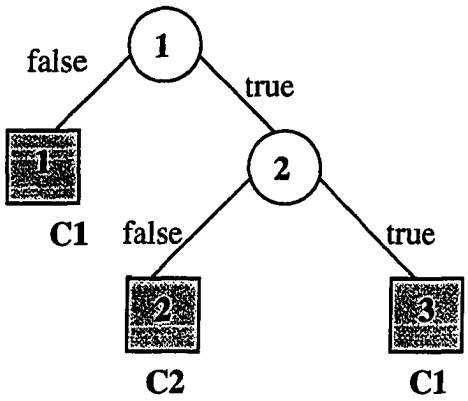

Figure 6 Example of Binary Decision Tree

The decision tree partitions the samples into three mutually exclusive groups, one for each terminal node. The rules can be expressed in disjunctive normal form: two rules cover class C1: $1' \vee (1 \wedge 2)$; one rule covers class C2: $1 \wedge 2'$, where 1' and 2' are negations of conditions 1 and 2 respectively.

The tree-based learning systems try to find a small decision tree, representing the set of training data. It is relatively easy to find a decision tree covering the training samples without errors, but in many cases the tree will make errors on new (test) data. The tree nodes are split into disjoint groups based on results of the test conducted in the node. The splitting is repeated for other nodes of the tree until all members of a sample belong to one class.

After every branch, decision about the next node to split must be made. Selecting nodes for a split randomly results in larger trees and is time-consuming. There are a number of methods used for choosing a node of the tree for a split, i.e. selecting a feature on which to split a node. The most widely used node-splitting technique aims at reducing the degree of randomness or "impurity" in the current node.

Often a sub-tree of a fully expanded tree (a tree covering all the decision rules in training data) performs better than the expanded tree. Pruning techniques allow finding an optimal size of the decision tree, i.e., they allow stopping splitting the nodes at the right time. The

pruning starts with a fully expanded tree. All the sample data is divided into training and test data. A tree is induced from training data and its performance (error rate) is checked on the test data. Starting from the bottom of the tree, the weakest branches are cut off. The idea is to find a sub-tree with lowest error rate.

3.5.3 Rule Induction

Inducing rules from sample data is similar to decision tree induction as every tree can be viewed as a set of decision rules. The restriction about the rules being mutually exclusive is relaxed when inducing rules. This can potentially result in more efficient and compact coverage of classes. If the rules are of the form: $x' \rightarrow C1$, $z \rightarrow C1$, $xy' \rightarrow C2$, it is possible that both rules corresponding to class $C1$ are satisfied at the same time. It is also possible that two rules can be satisfied for different classes. A main problem in building a learning system with a non-mutually exclusive set of rules is that it is not possible to predict how the rules will interact with each other even while training.

A search among all possible sets of rules in disjunctive normal form is very complex. The heuristic search procedure "predictive value maximization" finds the single decision rule of a fixed length, which maximizes the accuracy of the rule. In this approach, a relatively small set of promising expressions is kept in a table, and combinations of these expressions are used to produce longer expressions. The long expressions, covering the largest number of cases, are kept in a table and are in turn used for generating even longer expressions. The procedure is repeated until a single best rule, maximizing the accuracy of the rule for the class, of less or equal to a certain length n is generated.

First, for each variable, interesting threshold values are determined. Logical expressions with variables are generated. A small set of rules is kept in a table. It is then used to generate longer expressions. For example, one of the expressions in a table is: $b \wedge c$. If interesting thresholds for variable b are: $b > 10 \ \wedge \ b > 20$, and for variable c: $c < 30$, $c < 40$ then the expression $b \wedge c$ would lead to the possibilities:

$b > 10 \wedge c < 30, b > 10 \wedge c < 40, b > 20 \wedge c < 30, b > 20 \wedge c < 40.$

Longer expressions will be kept in the table instead of the original pruned expressions. After all interesting expressions, covering only cases of a particular class, are generated the best expression in the expression table is the answer.

Different techniques (such as resampling) are used to estimate error rates and to find the best length of the induced rule with minimum error rate. Longer decision rules will perform well on training data but performance on test cases may decrease. This happens due to over-specialization of the rule.

3.6 Measuring Semantic Similarity

Semantic similarity between two words is a quantitative measure of the degree to which the senses of the words are related. Semantic similarity is a widely used source of information for WSD. The following sections describe PMI-IR as a measure of semantic similarity and WordNet- based measures of semantic relatedness.

3.6.1 WSD by Web Mining for Word Co-occurrence Probabilities

In WSD by Web mining for word co-occurrence probabilities, Machine Learning is used to produce statistical models of training data and predict senses of the test data. The main novelty of this method of WSD is the way in which semantic features are generated. This method uses PMI-IR as a measure of semantic similarity when calculating semantic feature values. Word co-occurrence probabilities for the context words are gathered using the unlabeled data, collected by Information Retrieval (Turney, 2001).

Thus, the NRC WSD system uses two sources of information: a large unlabeled corpus for generating semantic feature values and a small labeled corpus for learning to classify

feature vectors. The process of semantic feature generation is unsupervised.

This method of finding semantic similarity does not depend on annotated corpora or a lexicon. Corpus-based approaches to WSD in the sections above, gather the information about the probability of co-occurrence of words from manually annotated corpora. The main shortcoming of those methods is data sparseness; manually sense-tagging is time consuming and expensive. PMI-IR addresses that problem by using Web or a collection of unlabeled texts: it "uses Pointwise Mutual Information to analyze statistical data collected by Information Retrieval" (Turney, 2001). PMI-IR as a measure of semantic similarity between two words is described in the next section.

3.6.1.1 PMI-IR as a Measure of Semantic Similarity

PMI-IR is an unsupervised learning algorithm for defining semantic similarity between two words. Semantic similarity indicates whether two words are semantically close to each other. For example, words such as *car* and *tire* are semantically similar, so they will have a high degree of semantic similarity whereas the meanings of words *car* and *printer* are not related to each other and these words are considered not semantically similar. PMI-IR uses Pointwise Mutual Information (PMI) to find semantic similarity between the words from the statistical data, obtained by Information Retrieval (IR):

$$\mathrm{PMI}(w_1, w_2) = \log_2(p(w_1 \wedge w_2)/(p(w_1)p(w_2))) \qquad (3.3.1)$$

In (3.3.1) $p(w1 \wedge w2)$ is the probability that words $w1$ and $w2$ co-occur. If the two words are statistically dependent, i.e. they tend to co-occur, then $p(w1 \wedge w2)$ is greater than $p(w1)p(w2)$ and the result of (3.3.1) is a positive number. If the two words are statistically independent then $p(w1 \wedge w2)$ is the same as $p(w1)p(w2)$ and PMI($w1, w2$) in this case has value zero. PMI($w1, w2$) can also be a negative number when $p(w1 \wedge w2)$ is less than $p(w1)p(w2)$, in which case it indicates that if one of the two words occurs the other one tends to be absent. Thus, the equation (3.3.1) explains mutual information between the words $w1$ and $w2$: if the two words are semantically related

then PMI will have a high positive value; when they are not semantically related, it will have a value of zero and when the two words are semantically irrelevant to each other (they tend not to have any semantic relation), the value of PMI will be negative.

PMI-IR uses Information Retrieval for calculating the probabilities in (3.3.1). For the NRC WSD system, the queries are issued to Waterloo MultiText System; for related work on synonym recognition (Turney, 2001) the Alta Vista search engine was used for obtaining the probabilistic information about co-occurrence of a pair of words.

There are different versions of PMI-IR (Turney (2001), Terra and Clarke, (2003)), depending on the kind of a query issued to a corpus or a search engine. The simplest query returns a number of documents, in which two words appear in the same document. This case will correspond to the document-oriented approach to estimating words' co-occurrence frequencies. The co-occurrence frequency of the two words will be the number of documents, where the two words co-occur, regardless of how far or close to each other, the words are located in the document.

Another case of document-oriented approach to estimating word co-occurrence frequencies is when there is a constraint about the distance (fixed-sized window) of co-occurrence. The frequency of co-occurrence of the words in this case is estimated by the number of windows where the two words co-occur within the document. The query returns the number of documents in which two words appear in the same document within the window of words. For instance, theWaterloo MultiText System allows the user to specify the window size. For calculating semantic feature values, the NRC WSD system used a 20-word window size.

Performance of the PMI-IR depends on the size of data (collection of documents), to which the queries are issued as well as the form of the query. For the task of synonym recognition for 80 synonym test questions from Test of English as a Foreign Language (TOEFL) and 50 synonym test questions from English Second Language (ESL), PMI-IR obtained a score of 74 %. The average score on TOEFL synonym test questions, obtained by ESL applicants to US colleges is 64.5%.

PMI-IR is a good estimator of semantic similarity (Turney, 2001). It overcomes limitations of lexicon-based similarity measures such as lack of links between different parts of speech, poor coverage of words in certain domains, inconsistencies of dictionaries and shortcomings of previous corpus-based methods such as data sparseness. PMI-IR exploits Web or a large corpus of unlabeled text for defining semantic similarity between a pair of words.

3.6.2 WordNet

WordNet (Miller,1995) is a freely available electronic dictionary, containing definitions for nouns, verbs, adjectives and adverbs[8]. It is a widely used lexical resource for WSD. Its design is based on psycholinguistic theories of how the human lexical memory works. It combines the features of dictionaries and thesauri; it includes definitions for individual senses of words as in a dictionary and it defines synonymy relations between words which are organized in a hierarchy as in a thesaurus, organizing related words into synonym sets or synsets. For example, the noun *mistake* has three meanings listed in WordNet 2.0:

Meaning 1: a wrong action attributable to bad judgment or ignorance or inattention. Synonyms: error, fault.

Meaning 2: an understanding of something that is not correct. Synonyms: misunderstanding, misapprehension.

Meaning 3: part of a statement that is not correct. Synonym: error.

In addition to providing information about synonymy, it also includes a variety of semantic relations such as hyponymy[9] (is-a), antonymy, meronymy (part-of), etc. This

[8] http://wordnet.princeton.edu/

[9] For example, *tulip* is a hyponym of *flower*, opposite relation is hypernymy: *plant* is a hypernym of *flower*.

organization creates a network in which semantically related concepts (senses) can be identified by their distance from each other.

The concepts are organized into hierarchies within a particular part of speech; they do not cross part of speech boundaries. For nouns "is-a" relation means that one concept "is-a-kind-of" another concept. This relation is referred to as hyponymy. For example, an *apple* is a hypernym of *fruit*. Similarly to nouns, verbs have a relation "is-way-of-doing" or troponymy. As an example, *talk* is a troponym of *communicate*. The top node element in each of these hierarchies is very general and the nodes on the lower levels are more specific. "Is-a" relation for nouns is the most developed relation in WordNet. Each hierarchy can be thought of as a tree, where the root node is the most general concept and the leaves are specific concepts.

Path lengths between the nodes (concepts) are not consistent in WordNet, as the concepts higher in a hierarchy are more general than the ones that are lower in the hierarchy. For instance, the path length between two general concepts can be the same as the path length between two specific concepts. In the case of general concepts, the path length will imply a big difference between the concepts, whereas the same path length between the specific concepts may not. An example from Pathwardan et al.(2003) illustrates the problem: in WordNet the concepts *mouse* and *rodent* are separated by a path of length one, which is the same distance that separates *fire iron* and *implement*. However, the concepts *fire iron* and *implement* are more general in WordNet (i.e., higher in the hierarchy) than the concepts *mouse* and *rodent*.

The fact that the distance or path length can be interpreted differently in WordNet depending on where in a hierarchy the concepts are located, has led to the development of a few measures of semantic relatedness based on the path length.

3.6.2.1 WordNet-based Measures of Semantic Relatedness

Semantic relatedness between concepts in WordNet covers semantic similarity and other

semantic relations such as hyponymy, meronymy, antonymy and others. WordNet-based measures of semantic relatedness assign a value that quantifies the degree to which two concepts are related to each other. Some WordNet-based measures of semantic relatedness such as *lch* (Leacock and Chodorow, 1998), *hso* (Hirst and St-Onge, 1998), *wup* (Wu and Palmer, 1994) use path length in WordNet to assign a value of semantic relatedness between a pair of words. Other measures such as *res* (Resnik, 1995), jcn (Jiang and Conrath, 1997), and *lin* (Lin, 1998) use both the path length and small corpora to define semantic relatedness between two words. All of those methods are based on concept hierarchies in WordNet.

The *random* measure generates random numbers as a measure of semantic relatedness of word senses. It is normally used as a baseline with which to compare the results of other measures of semantic relatedness. The *lch* measure finds the shortest path between two concepts. The shortest path between two concepts is the one that includes the fewest number of nodes (concepts). This value is then adjusted to take into account the maximum length in the taxonomy. The *lin* measure calculates semantic relatedness by using a formula from information theory. The *res* measure uses information content to find out relatedness between two concepts. The information content is a value that is assigned to each concept based on the information found in a corpus. The *jcn* measure uses information content defined by Resnik and also path length between concepts. The *wup* finds the path length to the root node from the most specific concept the two nodes share as an ancestor.

4 The Problem of Context

This chapter first describes the categories of context as presented in the literature on WSD. It then introduces studies of sense ambiguity and micro-context, followed by a description of systems performing WSD by using local and topical contexts. Topic context and domain information used for WSD are then discussed. The hypotheses of the thesis and the methodology for testing each hypothesis conclude the chapter.

4.1 How Much Context is Enough?

As already noted in Chapter 2, different concepts of context are distinguished in the literature on WSD: local context, topical context and domain information. However, the definitions of local and topical context (and respectively local and topical features used in Machine Learning) are not systematic.

According to one type of context categorization (Ide and Veronis, 1998), a micro-context is considered to be a small window of words surrounding the target word, from a few words to the whole sentence, in which the target word appears. Relation of the words with respect to the target word in micro-context can be considered (relational approach) or ignored (bag of words approach). An important parameter in analyzing micro-context in relational approaches is the distance with respect to the word to be disambiguated. Topical context includes substantive (content) words usually within a window of several sentences, which are typically used as a bag of words, in which words in the context are "regarded as an unordered set". Some methods use a window of 50 - 100 words.

Leacock et al. (1998) suggest that topical context can be represented by the words that occur anywhere in a window of context, while local contextual features are words that occur within close proximity to the target word. Crestan (2004) differentiates short and long contexts. The short context includes all the words in a small window of maximum three words. The long context uses "semantic clues" at a paragraph level. Similarly,

Paliouras et al. (2000) divide the context around a word to be disambiguated into two groups:

> *In local WSD only the close neighborhood of the word (<10 words on each side) is used. Topical methods on the other hand use a larger context window (> 50 words on each side).*

However, for example in Mihalcea (2004), local features among others include "the first verb before AW" (where AW is ambiguous word), "the first verb after AW", "the first noun before AW", and "the first noun after AW". Thus, the content words immediately preceding and following the target word are considered as local context.

In other systems, the topical and local contexts are distinguished, based on syntactic and semantic content of the ambiguous word. The local context is used for syntactic features and the topical context is used for semantic features. In Yarowsky (1994a) the topical and local contexts are distinguished as contexts being useful for semantic features and syntactic features respectively:

> *Semantic or topic-based ambiguities warrant a larger window (k ~ 20-50), while more local syntactic ambiguities warrant a smaller window (k ~ 3 or 4).*

In the NRC WSD system, Turney (2004) uses two types of context for generating syntactic and semantic features of the head word. Syntactic features are generated from local context (these are all words within a small window of -2 to +2 words around the head word). Semantic features are based on content words in a larger window of -4 to +4 words around the head word inside a sentence in which the head word appears, which can be considered as topical context. Thus, syntactic features use local context and semantic features use topical context.

Domain information is another source used for sense disambiguation, side by side with micro- and topical contexts. Taking into account the domain information, the target word's definitions are limited to the meanings appropriate in a particular domain. For

instance, the word *mouse* in computer science domain will have a meaning of an input device and not an animal. WSD by using domain information uses the context of the whole document where the ambiguous word occurs.

From the discussion above, we conclude that different concepts of context exist, and their definitions vary from author to author. In the following sections, we review a few studies of WSD and context, which will help us interpret the results of the experiments described in this thesis.

4.2 WSD and micro-context

An experimental study of the role of context for ambiguity resolution, conducted by Kaplan in 1950, used micro-context information to answer the question by Weaver (quotation in Chapter 1) about the optimal number of words, *N*, for the window of context words around a given ambiguous word. The objective of the experiments was to investigate such issues as the impact of the words, immediately preceding and following the target word, on WSD; the importance of semantic content of the context and some other issues such as accuracy of translators.

Experiments were conducted with a different window size around the target word. Seven translators were asked to translate (thus to disambiguate) the target words taking into account the various contexts; one or two words, preceding or following the target word, as well as the entire sentence.

In one group of the experiments, context words were allowed to be particles (e.g., stop words such as articles, prepositions, conjunctions). Experiments with two context words on either side in this group of experiments performed insignificantly better than when one context word on either side was taken into account.

In another group of experiments, more relevant to the experiments of this thesis, context words within a window consisted of at least one content word. The purpose of these

experiments was to take into account the semantic content of the context, compared to the first group of experiments, in which particle words contributed mostly syntactic information. These experiments were referred to as experiments with "substantive contexts". The results in this group of experiments demonstrated that the best performance is obtained when there are two context words (one preceding and one following), though the difference in the performance between one and two context words around the target word is not substantial. Performance of WSD when there are two context words on either side is reported to be not significantly different from when the entire sentence was considered.

The next experiments demonstrated a strong dependency of disambiguation results on the accuracy of translators, which was correlated with educational level, language skills and other subjective factors. To eliminate the 'human factor' from the process of disambiguation as much as possible, the following experiments were tested on the translators with similar accuracy results (according to the previous experiments). Substantive contexts were tested only on accurate translators. Disambiguation results showed that two context words (one preceding and one following) were not significantly better than four context words (two preceding and two following). Disambiguation based on the entire sentence context was not significantly better than two-word context centered on the target word.

As seen from the above, the results of the experimental study conducted by Kaplan suggest that the optimal window size of the micro-context is relatively small. Moreover, it is not significantly better or worse than the entire sentence, in which the ambiguous word is encountered. In case of the same level of translator's accuracy, the optimal number of context words is one word on each side, increased to two words if context words are particles.

Ide and Veronis (1998) describe the same phenomenon. From them we learn that there have been a few attempts by several researchers since the study by Kaplan to investigate this issue. The study conducted by Choueka and Lusignan (1985) focused on disambiguation by short context. A context is considered as a sequence of words of

"undefined length but bounded by sentence marks", preceding or following a given ambiguous word. Short context is referred to as a maximum of two words on each side of a word to be disambiguated. The objective of the study was to investigate if short contexts were sufficient for disambiguation in general and how short they could be. Experiments, similar to the ones conducted by Kaplan for English, were run by Choueka and Lusignan for French. Six experts with a Bachelor's degree in humanities, were asked to assign senses to ambiguous words based on the short contexts provided to them. The short contexts used in the study were: one word either preceding or following the target word or symmetric context (one word on both sides). Similarly, 2-contexts included two words either preceding or following the word to be disambiguated or two words on both sides.

The experts were initially asked to disambiguate relying on 1-contexts. If the context was not sufficient for disambiguation, then 2-contexts were used. The results of disambiguation demonstrated that in 9 out of 10 cases, 2-contexts were sufficient for disambiguation, but also 1-contexts were sufficient in almost 8 out of 10 cases.

An interesting observation has been made about using 2-contexts: error rate in WSD using 2-contexts was significantly higher than that of 1-contexts:

> *This is rather paradoxal, since 2-contexts should, for disambiguation purposes, be (intuitively) more reliable than 1-contexts (because they contain more information*).

The conclusion drawn by the authors of the study is that even though 2-contexts provide more information, there are also more "traps" in that information. The traps are the words that confuse the expert about which sense of the ambiguous word is appropriate. Wider contexts (located further from the target word) might contain more irrelevant information or "noise", which results in worse disambiguation results.

Ide and Veronis (1998) also mention the results of an experimental study by Yarowsky (1993, 1994a and b), according to which WSD in micro-context requires a window of 3- and/or 4-contexts. Yarowsky (1993) makes an interesting observation regarding the

disambiguation of different parts of speech: noun senses seem to be dependent on topical information, while verbs and adjectives are better disambiguated using local information. From the discussion above, a conclusion could be drawn that the window of context words required for success in WSD from local context is relatively small.

4.3 Local or topical context?

Crestan (2004) used semantic classes of the context of the target word for WSD by building Semantic Classification Trees (SCT). A semantic class corresponds to the most general node of a concept in WordNet, which is a hypernym to many concepts linked to it. For example, *plant* is a hypernym of *flower, tree,* and *bush.* Thus, the three words have the same hypernym (i.e., belong to the same semantic class).

The trees can be viewed as simple binary decision trees. Training data is used to build one or more trees for each word to be disambiguated. The nodes of the tree contain questions about the words in context of the word to be disambiguated. For example, if the target word *sense* is preceded by the word *make* than the question in one of the nodes of the tree will be: "-1#make?", where -1# shows the position of the word with respect to the target word *sense.* The end node of a branch of the tree contains the sense, which is assigned to the training case. The cases from the test data are presented to the tree and the target words are disambiguated based on their context, following either branch of the SCT. Different sizes of context were considered: short context (up to three words on each side of the word to be disambiguated) and long context (words from a paragraph level). For the short context experiments, the semantic classes of WordNet[10] were used, which allowed "generalization over words sharing the same high-level hypernym". The WordNet's top ontology consists of twenty-six semantic classes for the nouns and fifteen semantic classes for verbs. The information about semantic classes of training instances was included into the questions of SCT based on the training data. For the target words in the test data, semantic classes of the context words were obtained from WordNet and

[10] WordNet is presented in section 3.6.2.

compared to the ones in the SCT. If a context word in the test data and training data in corresponding positions shared the same semantic classes, the target word in the test data was assigned the sense of the training case.

For instance, suppose the verb *open* is followed by the word *box* in the training case and in the test case the verb *open* is followed by *jar* or *business*. Semantic classes of the words {box and jar} or {box and business} are compared with each other:

box	*06*[11]	20	*23*	25	35
jar	*06*	11	*23*	38	42
business	04	09	14		

Comparing the semantic classes of the words *box* and *jar,* it is found that they have two semantic classes in common (06: nouns denoting man-made objects; and 23: nouns denoting quantities and units of measure), whereas *box* and *business* do not have any semantic classes in common. The first case will be disambiguated, i.e., the verb *open* will be assigned the class of the training data: 'cause to open or to become open'. The second case would correspond to the sense of open as 'start to operate or function or cause to start operating or functioning' and would be disambiguated based on the training example (similarly by using the semantic classes of context) in which the target word is assigned to that class of the word *open.*

For the long context experiments, the large-scale semantic dictionary, developed by Crestan (2004), was used to obtain information similar to WordNet's semantic classes. All the words of the language are organized into 800 semantic dimensions. For example, the word *diary* is present in the dimensions: *calendar, story, book, and newspaper.* The context of the whole paragraph was used to obtain the most representative dimensions in the training data. That information was included in the SCT. This method used context on a wider scale than the one used with short context SCT.

[11] The numbers represent semantic classes (lexicographer files) in WordNet.

The results of disambiguation by short and wide contexts on the data of Senseval-2 ELS task demonstrated that some words benefited from the wider context (such words as *dyke, spade*) while the others had a significant decrease in the score (such as *authority, post*). The hypothesis drawn by Crestan (2004) is:

> *...the use of wide context semantics is mostly benefic in the case of homonymy, while it is not when dealing with polysemy.*

In a similar study on using semantic classes for WSD (Kohomban, 2005), there is an observation about the context size necessary for WSD when using semantic classes:

> *As we understood from our initial experiments, wide-window context features and topical context were not of much use for learning semantic classes from a multi-word training data set. Instead of generalizing, wider context windows add to noise, as seen from validation experiments with held-out data.*

The local context features and part-of-speech features were used in the study with the window of words varying from one to three words on each side of the target word. The window did not exceed the boundaries of the sentence. The results showed that "a window of two words to both sides yields the best performance for both local context and POS [part of speech] features" (Kohomban, 2005).

Leacock et al. (1998) attempted to assess the role of topical context versus local context. Their conclusion was that, for a statistical classifier, micro-context contributes more to sense resolution than the topical context.

In Agirre and Martinez (2000), the role of context is investigated for generating features for supervised Machine Learning. They analyzed the performance of topical features versus local features. They considered bigrams and trigrams (part-of-speech tags and the word) as local features. A bigram consists of two contiguous words (e.g. 'interest in'), a trigram, of three contiguous words. For topical features, they used all the words in the sentence plus a four-word window around the sentence of the target word. The

experiments used the SemCor[12] (Fellbaum, 1997) and DSO[13] (Ng and Lee, 1996) corpora. For the experiments with SemCor, the topical context performed well for nouns but not other parts of speech (verbs, adjectives, adverbs). For the DSO corpus, the results are in contradiction with the ones from Semcor: local features perform better for both nouns and verbs. Another observation is that the combination of all features (local and topical) results in lower precision (precision is described in Chapter 3) compared with the case when only local features were used for disambiguation. The authors of the research conclude, regarding the important features for WSD:

> *...the basic set of features is enough. Larger contexts than the sentence do not provide much information, and introduce noise. [...] Local features, i.e., collocations, are the strongest kind of features...*

Lamjiri et al. (2004) investigated the role of context size necessary for WSD. They consider two contexts around the target word: large window and sub-window centered on the target word. All words within the sub-window are taken into account, while the large window contains only content words. The system decides on the optimal window and sub-window sizes for every word to be disambiguated "by examining possible window size values ranging from 25 to 750 characters" for sub-window and window sizes respectively. The system uses a somewhat unusual method of estimating the context size; instead of words, the characters are counted, making sure words are not cut at the extremities. For example, for the verb *add,* the system found the window size of 100 characters and sub-window of 25 characters to be optimal; for the adjective *simple,* the corresponding numbers of characters are 400 and 25 respectively. The authors conclude that the window size is related to the distribution of the senses in the training samples available for a word:

> *...a large window size is selected when the number of samples is large, and the samples are not evenly distributed among senses.*

---

[12] SemCor is a corpus containing 360,000 words in which all the content words are sense-tagged with WordNet senses.

[13] The corpus includes 191 polysemous nouns and verbs. For each of the words there are 1000 sentences, in which only the target word is sense-labeled with WordNet senses.

It is necessary to note that the system considers 'large context' for both window and sub-window sizes; for some words to be disambiguated the sizes of the two windows need to be larger than for the other words.

4.4 Topic signatures for WSD

A study by Agirre et al. (2001) aimed at enriching WordNet concepts with topic information (topic signatures) to overcome the lack of topical links among concepts in WordNet. Every concept in WordNet was associated with a topic signature (a set of related words), which was constructed from the web or a sense-tagged corpus. For instance, the noun *waiter* has two word senses in WordNet:

1. *waiter, server* – a person whose occupation is to serve at table (as in a restaurant);

2. *waiter* – a person who waits or awaits.

For each of these concepts a list of semantically related words is obtained either by querying the web or from a sense-tagged corpus. For example, the lists could contain the following words for the two senses of *waiter*:

1. restaurant, menu, waitress, dinner, lunch, etc.

2. hospital, station, airport, boyfriend, girlfriend, etc.

The queries are built using the information in WordNet (similar to Mihalcea and Moldovan, 1999). The queries are composed of the synonyms of the words, defining phrases from the word definitions and their different combinations using AND and NEAR operators. Every concept is assigned a collection of documents, which is then processed to find the semantic cues (most frequently used words in the document collection in some context) that are helpful for deciding on the topic of the word. The topic signatures were evaluated on a WSD task. The idea of the algorithm for WSD is: given an occurrence of an ambiguous word in the text, the words from the context of the

target word are collected and are compared with the topic signatures of the target word (each sense of the target word has its corresponding topic signature). The sense of the target word, which has the highest number of words from the context occurring in the signature, wins. Different context sizes were tested in the experiments: a large context (100-word window) and a sentence context (words limited by sentence boundaries).

The results of experiments demonstrated that topic signatures are a useful source of information for WSD (however, it is a complementary source of information to the local collocations and other local features). The experiments consisted of two parts: in the first part the topic signatures were constructed by querying the web and in the second part, they were constructed using the sense-tagged corpus (SemCor). In both experiments, the results of disambiguation in the case of the sentence context proved to be better than the results of the experiments with the large contexts.

Another observation of the authors of the paper, which regards the context size, is that when querying the Web, the retrieved documents "introduce a certain amount of noise into signatures". One of the important methods to reduce the noise is to "filter" the signatures by limiting the context of the words extracted from the retrieved documents to the sentence. Thus, according to the research, the sentence contains more relevant semantic cues, which are then incorporated into the topic signatures.

From the above, we conclude that the sentence where the ambiguous word occurs is a valuable source of information for WSD. Even though the focus of the research was not to investigate the role of different contexts for obtaining the topical information (which is expressed in the topic signatures corresponding to each concept of the ambiguous word), the results of the experiments demonstrated that the sentence context contains more useful information than larger contexts of 100 words, for deciding on the topic of the ambiguous word.

4.5 Domain-driven WSD

A study by Magnini et al. (2002) investigated the role of domain information in WSD. The extended version of WordNet, WordNet Domains, was developed by Magnini et al.(2002), in which for every synset of WordNet there is a corresponding domain annotation (such as Medicine, Architecture, Sport). About 200 domain labels were used to construct WordNet Domains. Synsets may be assigned one or more domain labels. Domains group together words from different syntactic categories. For example, Medicine domain, groups together senses from nouns, such as *doctor#1*[14] and *hospital#1* and from verbs, such as *operate#7.* Some of the synsets of WordNet could not be assigned any domain label as they do not belong to a certain domain but rather can appear in almost all of them. For this reason, a Factotum label was created. It includes two types of synsets: generic synsets, which are hard to classify with a particular domain, such as the word *man#1* (i.e. an adult male person); and stop sense synsets, which appear in different contexts, such as *numbers, days, weeks, colours.*

For the purpose of WSD, forty-three domain labels were used. For the WSD task, an observation from domain-oriented preliminary text analysis was taken into account: "relevant domain" actually makes sense within a portion of text (i.e. a context), rather than with respect to the whole text". This approach allows dealing with WSD taking into account possible variation of domains within a given text. The general idea underlying the WSD algorithm implemented in the research is that sense disambiguation of a word in its context is a process of "comparison between the domain of the context and the domains of the senses for such word". Domain information is represented by a domain vector, which has a length of the number of considered domains (43 domain labels from WordNet Domains). Two kinds of domain vectors are distinguished: text vectors and sense vectors. Text vectors are computed according to the senses of the words in the context, while the sense vectors are induced by the system by using the information from WordNet Domains (or exploiting training examples previously labeled with WordNet Domains labels).

[14] *doctor#1* corresponds to the sense number one of the word *doctor* in WordNet.

First, the text vector is computed by selecting a context of 100 content words centered on the target word (the tests on SemCor showed that the context of fewer than 100 content words decreased the system's performance). Then the domain annotations from WordNet Domains for all the synsets of the selected words are collected and the frequency of each domain in this set is computed. The domain labels for each sense of the target word are then compared to the information about domains of the context. The system chooses the sense of the target word, which maximizes the similarity with the text vector.

The ITC-irst system (Magnini et al., 2002) participated in Senseval-2 'all words' and ELS tasks. It accomplishes WSD without other syntactic or semantic information except domain labels. The results of the system's performance suggest that domain information is a useful source for WSD. 'All words' task benefited from the domain-driven WSD more than ELS task as the texts were long enough to provide an accurate context of 100 content words around the target word (whereas in ELS task the contexts were shorter than 100 content words). One of the conclusions from the results is that words, which were labeled with Factotum domain label in WordNet (such as all the senses of the word *begin*), lie outside the domain approach to WSD and could be disambiguated using local information.

Agirre and Martinez (2004) use a rich set of features for the Senseval-3 ELS task. Among the features are the domain features, which use the labels in WordNet Domains to identify the most relevant domain in context.

To conclude, domain labels are useful for WSD as they establish semantic relations among word senses that are used during the disambiguation process. Domain-driven WSD relies on a large context of content words around the word to be disambiguated.

4.6 Motivations for Experiments

From the studies on WSD and context discussed in the previous sections, we observe that some of the work seems to suggest that a wider context is not useful, whereas other

research seems to suggest that a wider context is beneficial for WSD. It is important, however, to notice that the former are dealing with a local/syntactic/micro context. And the latter with a topical/domain/semantic context.

Thus, it seems that a wider context can be useful if it is treated semantically (by exploiting the meaning of the context words), but is not useful if it is treated syntactically (by using only the surface form of the context words).

We are interested in experimenting with a wider context by taking into account more content words from the context of an ambiguous word. Thus, our objective is to add more semantic content to the words to be disambiguated and observe how this will affect the WSD results. The initial hypothesis we make in the light of our objective is restated in the next section and is followed by our other hypotheses.

4.7 Hypotheses

We hypothesize that a bigger context window will be useful in WSD. A bigger context window for our experiments contains only content words, thus we are considering a bigger window for the *semantic* features in the NRC WSD system, not for the *syntactic* features in the NRC WSD system.

Hypothesis 1: "The accuracy of the modified version of the NRC WSD system will improve as more *semantic* features are introduced, by expanding the window of *content* words around the head word."

The modified NRC WSD system was evaluated using a different window of context words around the word to be disambiguated. The idea of the first hypothesis was that the more content words around the head word are taken into account for WSD, the better should be the chances for disambiguation due to more available semantic content. However, the results of the experiments with different window sizes around the word to be disambiguated have demonstrated that the system's accuracy did not improve when a

larger number of words was taken into account for WSD compared to the case when only the two nearest content words from the context were taken into account.

From these results, we formulated the second hypothesis of this thesis.

Hypothesis 2: "Introducing additional semantic features in the modified NRC WSD system, by expanding the window of content words around the head word, did not improve accuracy because the new features were irrelevant for disambiguating the head word."

The second hypothesis aimed at investigating whether the distant content words were less relevant than the two nearest content words. In this case, disambiguating the head word would not benefit from taking into account the two distant words. An alternative hypothesis is that the new features were relevant but redundant. That is, the new features had relevant information, but the information was redundant, because it was already contained in the content words that are closest to the head word. The reason, we repeat, for comparing the performance with the two nearest words versus the two furthest words is that the comparison can distinguish these two hypotheses. If the information is redundant but not irrelevant, then it should not matter whether we use the two nearest or two furthest words. If the information is irrelevant, then it will matter, and the furthest words will not work as well as the nearest words.

Our experiments rejected Hypothesis 1 and confirmed Hypothesis 2. We wondered how general our results were. Perhaps they were specific to the NRC WSD system; in particular, perhaps they were specific to PMI-IR. Therefore, we decided to test their generality by substituting another class of measures of semantic similarity in place of PMI-IR.

Hypothesis 3: "If the NRC WSD system is modified by replacing the statistical measure of semantic similarity (PMI-IR) with lexical measures of similarity (using WordNet), Hypothesis 1 will still be rejected and Hypothesis 2 will still be accepted."

Thus, we hypothesized that the general pattern of the results in experiments when semantic feature values are calculated using PMI-IR and WordNet-based measures of semantic relatedness should be similar.

The results of the experiments will be presented in Chapter 5. The following section presents the modifications made to the NRC WSD system with the purpose of testing the hypotheses.

4.8 Methodology for Testing the Hypotheses

Following the idea of our *first hypothesis*, we were interested in extracting more context words (semantic features) around the head word, which would potentially provide more useful information (semantic content) and contribute to the WSD process. Only semantic features were modified in the NRC WSD system; syntactic features remained unmodified.

The objective of the modifications to the NRC WSD system was to experiment with a different number of context words around the word to be disambiguated. We started with four context words[15] and continued the experiments; every time eliminating one context word. A special case of experiments was the one with two semantic features, consisting of two parts: experiments with the two nearest from the head word features and the two furthest from the head word features. For completeness, we also tried the experiment without semantic features, that is, with only syntactic features.

Figure 7 shows an example from the training data of the ELS task of Senseval-2:

> When things are on the up and the lodestar of a transformatory politics shines bright, so too does the avant-garde project of overcoming the separation of art and life (p. 171). In this perspective it seems that Callinicos can only mean

[15]The more context words we use, the slower the system runs. That is why we did not go beyond four context words in our experiments.

relatively little with his disclaimers about good art. The individual good work might get thrown up, however unpropitious the circumstance. But it can only be a quirk; and the force of its goodness; is strictly limited and circumscribed. Only once, in a fleeting reference to Matisse is there a sense of the boot being on the other foot, of art offering a sense of liberation from social ideology. But even this is done in the name of a supposed immediate sensuous charge rather than any more extended critical capacity of **<head>*art*</head>**or the aesthetic.

Figure 7 An example extracted from the training data of the ELS task of Senseval-2.

After the examples of the training data are part-of–speech tagged, all the nouns, verbs and adjectives are extracted from the training example, omitting the words with other parts of speech. For every head word in examples, three content words are extracted to the left of the head word and one content word, to the right of the head word. For our example, the words are:

extended/VBN critical/JJ capacity/NN aesthetic/NN.

The original NRC WSD system had the constraint of not crossing the sentence boundaries. That constraint conflicts with our desire to try a wider context window, therefore we had to abandon it. Consequently, the search for content words continues until either all the positions are filled or the end of the example is reached. If words are missing for some positions, then special null characters are inserted in place of the missing words. As already stated, the choice of an asymmetrical window around the head word was caused by the format of Senseval-2 ELS task's data: as a rule, the head words were located at the end of the last sentence of the paragraph.

In the case when four context words are extracted around the head word, semantic feature names of the form *position_model* have four different positions: *pre1* (preceding1), *pre2* (preceding2), *pre3* (preceding3) and *fol* (following): *pre1 pre2 pre3* <HEAD> *fol.*

For the example above the extracted features will be:

pre1_extended pre_2critical pre3_capacity fol_aesthetic.

Model is a noun, verb or adjective, which is extracted from the training data. For example, if the nearest noun, verb or adjective, preceding the head word is "abstract", then the feature name will be *pre3_abstract.* Similarly, for the experiment with three content words, the features were of the form *pre2_model, pre_3model* and *fol_model.* Three semantic features extracted from the training example will be:

pre2_critical, pre3_capacity, fol_aesthetic.

The semantic features with the name: *avg_position_sense* are generated in a similar way with the original version of the NRC WSD system. *Position* is one of the four possible positions with respect to the head word and the *sense* is any label of the sense, in which the head word is used in the training data. Therefore, there will be three average features for the preceding features (one for each position) and one, for the following, when four words are taken into account for WSD.

The values of semantic features *position_model* and *avg_position_sense* are calculated in a way similar to the original version of the NRC WSD (described in Chapter 3) system for the experiments, using PMI-IR as a measure of semantic similarity. Semantic features with the names *position_model* are normalized by being converted to percentiles. The *pre1, pre2, pre3* and *fol* features are normalized separately. That is, each feature vector is normalized internally (according to its own values) and not externally (taking into account values of other feature vectors).

For testing the *second hypothesis,* the objective of the experiments was to find out why adding more features did not cause the accuracy to improve. Because the best performance (taking into account the accuracy and speed) was obtained in the case of two features, which were the closest to the head word, we tried the experiment with only two additional features in positions *pre1* and *pre2.* The semantic features had the form: *pre1_model, pre2_model.*

For testing the *third hypothesis,* WordNet-based measures of semantic relatedness were used in place of PMI-IR for calculating semantic feature values. The experiments with six WordNet-based measures of semantic similarity followed the same scheme, with

different numbers of context words around the head word, as in the case of experiments using PMI-IR. Experiments also included the special case with the two furthest context words. In the implementation, we used the freely available WordNet::Similarity package, implemented by Pedersen et al. (2004), to calculate sense similarities.[16] For the experiments, the similarity between two words is defined mainly by finding the distance between the two words in WordNet's topology (Some measures also use a small corpus for defining the similarity, as described in Chapter 3). The distance and similarity are inversely correlated.

[16] http://wnsimilaritysourceforge.net.

5 Experiments

This chapter presents the score and statistical significance tests' results of the experiments for testing Hypothesis 1, Hypothesis 2 and Hypothesis 3. Discussion of results with a summary of all the experiments and comparison of the results of our study with related studies, conclude the chapter.

As already mentioned, the experiments with modified versions of the NRC WSD system, consisted of experiments with a different window of context words surrounding the head word in the ELS task's data. A subset of Senseval-2's ELS (30 words: 10 nouns, 10 adjectives and 10 verbs) was used for all the experiments. In both sets of experiments (based on PMI-IR and WordNet similarity measures), modified versions of the NRC WSD system considered the following number and position of semantic features:

- four semantic features (three content words before the head word and one after);
- three semantic features (two content words before the head word and one after);
- two nearest semantic features (one content word immediately before the head word and one immediately after);
- two furthest semantic features (two furthest content words from 4- word window);
- one semantic feature (one content word before the head word);
- no semantic features, only syntactic features.

For the experiments for testing the Hypothesis 1 and Hypothesis 2, we use the Fisher Exact Test (Agresti, 1990) for defining statistical significance of the experiments' results. The results of the test (for p value for the same or a stronger association) will be shown together with the results of the experiments above. The test is used to determine whether the difference in the score of two experiments is statistically significant. The Fisher test calculates the probability p of the difference between the data observed and the data expected. The threshold value of p for 95% significant result is 0.05. If the p value for the same or stronger association is < 0.05, then the result is considered statistically significant. Otherwise, the result is not statistically significant.

Tables with score information and statistical significance tests' results for each

experiment contain two numbers which correspond to fine-grained score (f) and coarse-grained score (c). The two scores are obtained based on WordNet's fine-grained and coarse-grained senses respectively. The coarse-grained senses, as already mentioned, were created by the Senseval-2 organizers, who manually organized the WordNet senses (for the head words only) into clusters of similar senses.

For testing whether the difference in the score of an experiment is significant, we compare the score of that experiment with the score of the corresponding experiment with the two nearest semantic features. The asterisk next to the score in the tables for testing Hypothesis 1 and Hypothesis 2 indicates that the score is significantly lower than the score of the experiment with the two nearest semantic features.

For the experiments for testing the Hypothesis 3, we use the Signs Test (Agresti,1990) instead of Fisher Exact Test.

Discussion of all the experiments' results is presented in Section 5.4.

5.1 Experiments for testing the Hypothesis 1

This section presents the results of experiments for testing the first hypothesis. Table 4 contains the score information and statistical tests' results for the experiments with different numbers of context words around the head word (semantic features were calculated using PMI-IR). For testing the first hypothesis, the experiments considered a window of content words ranging from zero to four. We compare the results of the other experiments (with four, three, one and zero semantic features) with the results of the experiments with the two nearest semantic features (2n in the table). The detailed results of the experiments and statistical significance tests are presented in Appendices A and B respectively.

	PMI-IR	
	f	c

4	65.1	69.9
3	65.5	70.5
2n	**66.3**	**70.8**
1	62.7*	67.7*
0	58.5*	63.4*

Table 4 Score and statistical significance tests of the experiments for testing the Hypothesis 1.

According to the results of the statistical significance tests, the system's score in the case of two nearest semantic features is:

- not significantly better than in the case of four or three semantic features;

- significantly better than in the case of one and zero semantic features.

The score obtained by the original version of the NRC WSD system with two semantic features for the same subset of 30 words (extracted from the table of all words' result) is slightly different from the score obtained by our experiments with two nearest semantic features: 67.2 and 71.7 for fine-grained and coarse-grained senses respectively. The results of our experiments with the two nearest semantic features are 66.3 and 70.8. The difference in the score could be explained by the fact that the sentence boundaries are treated differently in the original version of the NRC WSD system and in our modified version of the system. Another factor, which might have affected the results, is that the Waterloo MultiText System has been updated since the time of preliminary experiments of the NRC WSD system with Senseval-2's ELS task data. Our experiments are more recent than the preliminary experiments by the author of the system.

Thus, the best performance taking into account the accuracy and the speed, for the experiments using PMI-IR for calculating semantic feature values occurs in the case of the two nearest semantic features: three and four semantic features take a longer time to calculate and do not improve the accuracy; in the case of one or zero semantic features the system's accuracy decreases significantly.

5.2 Experiments for testing the Hypothesis 2

For testing the second hypothesis, we experimented with the two furthest content words (from the four-word window). Statistical significance tests shown in Table 5 demonstrate that the score of the system with the two nearest semantic features is significantly higher than in the case of the two furthest semantic features. Thus, the two furthest words are less relevant than the two nearest words for disambiguating the head word.

	PMI-IR	
	f	c
2n	**66.3**	**70.8**
2f	61.6*	66.4*

Table 5 Score and statistical significance tests of the experiments for testing the Hypothesis 2.

5.3 Experiments for testing the Hypothesis 3

For testing the third hypothesis, WordNet's measures of semantic relatedness (the measures are introduced in Section 3.6.2.1) were used for calculating semantic feature values in our system. For our experiments, we selected measures of semantic similarity from Pedersen et al. (2004) WordNet::Similarity software package, based on their speed. Due to the large number of similarity measurements required to calculate features for modified versions of the NRC WSD system, we required measures that could compare two words in less than (roughly) two seconds. The measures that met this requirement were *random, lch, lin, res, jcn, wup.* The *random* measure is used as a baseline for the following experiments using WordNet similarity measures.

The scores of the experiments are presented in Table 6.

	random		lch		lin		res		jcn		wup	
	f	c	f	c	f	c	f	c	f	c	f	c
4	59.4	64.1	60.9	65.3	60.5	64.8	61.1	65.3	60.6	64.6	61.4	65.9
3	60.5	64.8	61.2	65.6	62.0	66.7	62.0	66.5	60.4	64.9	61.7	66.2
2n	**60.7**	**65.4**	**61.7**	**66.1**	**63.4**	**68.1**	**62.1**	**66.8**	**62.1**	**66.7**	**62.3**	**66.9**
2f	58.0	62.7	60.1	64.5	59.0	63.6	59.4	64.0	59.6	64.0	59.6	64.0
1	60.0	64.6	61.4	65.7	60.8	65.8	61.5	66.1	61.2	66.2	61.7	66.4
0	58.5	63.4	58.5	63.4	58.5	63.4	58.5	63.4	58.5	63.4	58.5	63.4

Table 6 Score of the experiments for testing the Hypothesis 3.

To test the significance of the experiments' results, we repeat, we use the Signs Test. None of the WordNet's measures of semantic relatedness taken individually performed significantly better in our experiments than *random* measure, according to the Fisher Exact Test. Therefore, we did not consider them individually; instead, we combined their results, using the Signs Test.

To test whether the first part of the Hypothesis 3 is supported (i.e., whether for WordNet-based experiments, the Hypothesis 1 of the thesis is rejected), for each of the WordNet measures, we calculate the differences between the score of experiments with four semantic features minus the score of experiments with the two nearest semantic features. There are five measures (excluding *random*) with fine and coarse grained scores for each measure, yielding 10 scores for experiments with four semantic features and 10 scores for the experiments with the two nearest features. The 10 differences are all negative. That is, the score of experiments with the two nearest semantic features is always higher than the score of experiments with four semantic features. According to the Signs Test, the probability of this happening by chance is $p = 0.00097$ (1 over 2 to the power of 10), which is significant at the 95% level. Therefore, we can conclude that the results of the experiments with the two nearest features are significantly better than the results of the experiments with four semantic features, for the WordNet-based measures considered as a group. Thus, the first part of the Hypothesis 3 is supported.

To test whether the second part of the Hypothesis 3 is supported, we verify that Hypothesis 2 of the thesis should be supported (i.e., scores of the experiments with the two nearest semantic features should be significantly better than the scores of the experiments with the two furthest semantic features). For the Signs Test, for each of the WordNet's measures of semantic relatedness, we calculate the differences: scores of the experiments with the two nearest semantic features minus scores of the experiments with the two furthest features. Again, the score of the experiments with the two nearest semantic features is always higher than the score of the experiments with the two furthest semantic features. The 10 differences are all positive. According to the Signs Test, the probability of this happening by chance is $p = 0.00097$ (1 over 2 to the power of 10), which is significant at the 95% level. Therefore, we can conclude that the scores of the experiments with the two nearest semantic features are significantly better than the scores of the experiments with the two furthest semantic features, for WordNet-based measures of semantic relatedness considered as a group. This demonstrates that the second part of the Hypothesis 3 is supported.

Thus, the WordNet-based measures of semantic relatedness considered as a group, reject Hypothesis 1 and accept Hypothesis 2, although none of these measures is significantly different from *random* measure, when considered individually. This implies that the Hypothesis 3 of the thesis is supported.

5.4 Discussion of Results

Table 7 is a summary of the experiments' results. In the experiments with a corpus-based measure of semantic similarity (PMI-IR) and lexicon-based measures of semantic relatedness (*random, lch, lin, res, jcn, wup*) the two nearest features have the highest score. The table presents scores of all the experiments in fine (f) and coarse (c) grained senses. The numbers on the left correspond to the number of context words used in each

experiment (4, 3, 2n, 2f, 1) [17].

	PMI-IR		random		lch		lin		res		jcn		wup	
	f	c	f	c	f	c	f	c	f	c	f	c	f	c
4	65.1	69.9	59.4	64.1	60.9	65.3	60.5	64.8	61.1	65.3	60.6	64.6	61.4	65.9
3	65.5	70.5	60.5	64.8	61.2	65.6	62.0	66.7	62.0	66.5	60.4	64.9	61.7	66.2
2n	**66.3**	**70.8**	**60.7**	**65.4**	**61.7**	**66.1**	**63.4**	**68.1**	**62.1**	**66.8**	**62.1**	**66.7**	**62.3**	**66.9**
2f	61.6	66.4	58.0	62.7	60.1	64.5	59.0	63.6	59.4	64.0	59.6	64.0	59.6	64.0
1	62.7	67.7	60.0	64.6	61.4	65.7	60.8	65.8	61.5	66.1	61.2	66.2	61.7	66.4

Table 7 Summary of the experiments' results.

According to the Signs Test, this result is not due to a random chance: for each of the similarity measures we calculate the difference between the score with two nearest features (2n from the table) and the rest of the cases: four, three, two furthest and one (2n-4, 2n-3, 2n-2f, 2n-1). For the Signs Test, as already mentioned, we are interested in the sign of the difference. For each of our experiments (PMI-IR, random, lch, lin, res, jcn, wup), that difference in score is always positive. Thus, for each case of the experiments (1, 2f, 3, 4) we have seven positive values (for each of PMI-IR, random, lch, lin, res, jcn, wup). The probability of this happening by chance is 0.00781 (1 over 2 to the power of 7). The 95% confidence threshold is 0.05 and 0.00781 < 0.05, so we can be 95% confident that the best score for the experiments with the two nearest features is not due to random chance.

Thus, our initial hypothesis, that taking into account more content words around the word to be disambiguated would result in better performance, is not supported. For our experiments, a window of two context words (nouns, verbs or adjectives immediately preceding and following the target word) proves to be the most successful.

Trying to investigate what caused our first hypothesis to fail, we hypothesized that the reason for the decrease in performance could be that the additional features (in case of four semantic features the additional features will be in positions *pre1* and *pre2*) are less

[17] 2n and 2f correspond to 2 nearest and 2 furthest context words from the 4-word window.

relevant to the head word. (Again, an alternative hypothesis was that the additional features are redundant, i.e. they do not add any useful information to the information contributed by the two nearest features.) To demonstrate the idea behind the hypothesis, we show examples of the words extracted from training cases for the head word *art* (the four words are the three words preceding and one word following the head word):

summer piazza front gallery

extended critical capacity aesthetic.

The experiments with the two semantic features that are *furthest* from the head word demonstrated that the system's performance decreases significantly in this case compared to the experiments with the two *nearest* semantic features. In the examples above, the two nearest words are {front, gallery} and {capacity, aesthetic}. This supports the hypothesis that the distant words are irrelevant to the head word. If the words were redundant, then the score of the experiments with the two furthest semantic features would have been the same as in the case with the two nearest semantic features.

Finally, we hypothesized that we should get similar results for both sets of experiments: corpus-based and lexicon-based. This implied that the Hypothesis 1 would be rejected and Hypothesis 2 would be accepted for the experiments with WordNet-based measures of semantic relatedness. Statistical significance tests demonstrated that the Hypothesis 3 is supported for the experiments with WordNet's measures of semantic relatedness, considered as a group. To conclude, the experiments for testing Hypothesis 3 demonstrated that the results of our experiments are not specific to a certain method of defining semantic similarity between words. They follow the same pattern for both: corpus-based and lexicon-based measures of semantic similarity.

The observation from the results of the experiments is that by expanding the window of words around the head word, the chance of getting "noise" (i.e., irrelevant information) when disambiguating the word, is increasing. As discussed in the section on Machine Learning, when irrelevant features are present while training, the system's performance suffers because the model is over-fitted or over-specified to the training data.

In the next section, we attempt to compare the results of our experiments with the related studies on WSD and context.

5.4.1 Comparison of Results with Related Studies

The modified versions of the NRC WSD system take into account the context words in close proximity to the target word for syntactic features. Thus, syntactic features use local context. The context used for semantic features is difficult to categorize according to the schemes used by the systems presented in Chapter 4. On the one hand, the context used for generating semantic features considers only content words similar to typical systems using topical context, whereas in local context all the words within a small window are considered. Sentence boundaries are crossed in order to search for content words around the target word. This is typical to systems that use larger than a sentence context for topical features. Local features used in a typical system using topical context are collocations. The semantic features used in the NRC WSD and the modified versions of the NRC WSD are not collocations. These points would allow us to consider the context used for semantic feature generation as topical.

On the other hand, in 46% of all the training cases and 50% of the test cases for the 30 words used in the experiments of this thesis, the search for the content words went beyond the sentence where it was used. These statistics suggest that the context used for generating the semantic features cannot be considered as purely topical. In a typical system using topical context, a wider window of a few sentences is used and in our experiments, in less than half of the cases the context was wider than a sentence. Another point is that systems using topical context consider the context words as a bag of words, where the words taken from a context of a few sentences do not follow any order with respect to position, distance and other relations to the target word. In our system, the content words are fixed to a specific position (they can be the first, second or third content word, preceding, or first content word, following the target word). Another difficulty for counting the semantic features as topical, is that only four words around the

head word are taken into account compared to a greater number of words used in other systems. We believe that this cannot contribute significantly to deciding on the topic of the target word.

Different interpretations of local and topical context by recent systems may suggest that it might be more useful to "regard the two as lying along a continuum and consider the role and importance of contextual information as a function of distance from the target" (Ide and Veronis, 1998). The idea seems applicable for our study; that is, for our experiments we distinguish two contexts: short and wide, which are defined by the distance form the target word.

Despite the differences in context categorization, a general trend that can be observed from the discussion in Chapter 4, is that contexts larger than a sentence introduce noise, i.e. the words located from the target word further than within a sentence are generally irrelevant to the target word. They do not contribute to the target word's disambiguation. The study on enriching WordNet by using topic signatures, demonstrates that the context of the sentence provides more useful information than larger contexts of 100 words (Agirre et al., 2001). Similarly, the conclusion by Agirre and Martinez (2000) regarding local context versus topical context suggests that topical features (in a wide window of all the words within a sentence and a 4-word window around the sentence where the target word occurs) do not provide much useful information and add to noise.

The conclusions of the studies about local context on the other hand follow the common pattern: linguistic studies as well as studies by the systems performing WSD observe that the words in the immediate neighborhood of the target word are the most useful words for WSD in micro-context. Local collocations, bigrams, are found to contribute to WSD more than the other words within a sentence. They are found to be even more useful than topical features.

For modified versions of the NRC WSD system, we observe that the words in the close neighborhood (short context according to our definition) contain the most useful information for disambiguating the target word. The two distant words taken into account

(wide context: partially within the sentence of the target word and partially beyond the sentence) are less relevant to the target word than the two closest words.

The above-mentioned observations about the nature of semantic features suggest that the context used for generating the semantic features in the modified versions of the NRC WSD system, is neither purely topical nor purely local. We believe, this explains the system's performance in the experiments.

Expanding a window of context words beyond a sentence of the target word, did not result in a better accuracy of the system in our experiments. The results of our experiments demonstrated that a wide context did not contribute to disambiguating a target word more than a short context. Our experiments also demonstrated that the words in a wide context were less relevant (as opposed to being redundant) to the target word than the words in a short context. This is similar to the observations of related studies on topical context, which were presented in Chapter 4, such as studies by Agirre and Martinez (2000), Agirre et al. (2001), which emphasized the dominant role, for topical context, of the sentence of the target word over contexts that are wider than a sentence. For our experiments, the context was wider than a sentence of the target word nearly in half of all the training and test cases. Thus, this hypothesis that the context of the sentence of the target word is the most useful for topical context, could explain the results for the cases when the search for content words went beyond the sentence of a target word.

For the second half of the cases of the training and test data, when the context was limited to a sentence where the target word occurred, we believe it would be appropriate to attempt to understand the results taking into account the studies on micro-context. Our experiments demonstrated that the words in short context (the content words immediately preceding and following the target word) were most beneficial for WSD. This observation is similar to the observations discussed in studies on micro-context. Such studies demonstrated that the words in the close proximity to the target word within the sentence where it occurs, are the most beneficial for disambiguating the target word in micro-context.

Thus, we believe that the nature of the semantic features used in the modified versions of the NRC WSD system, reflected on the results of our experiments. The results of the experiments can be explained taking into account the dual nature of the semantic features used in the modified versions of the NRC WSD system.

Another possibility to explain the results of our experiments is to investigate whether wide contexts are useful when context features are based on properties of a group of words as opposed to the case when they are based on properties of a single word. It is possible that in the discussed studies, in which WSD benefited from wide context (such as studies on topic signatures and domains), wide context was useful not because context features were semantic or syntactic. It may be that wide context was useful because the context features were based on properties of groups of words as opposed to the case when context features are based on properties of individual words in the neighborhood of the head word. For example, in the study on topic signatures, each concept in WordNet had a corresponding topic signature, consisting of semantically relevant words. A target word was disambiguated by comparing its context with all the topic signatures of the ambiguous word. The signature that had a maximum overlap with the words from context won. Thus, the context features were based on a group of words (a group of words in a signature and a group of words from context). In the NRC WSD system, the context features are based on individual words. This hypothesis will have to be tested in future work.

5.4.2 Comparison of Performance: PMI-IR and WordNet Similarity Measures

As seen from the Table 7 in Section 5.4, PMI-IR based experiments outperformed all the WordNet-based experiments. PMI-IR has a number of advantages compared with lexicon-based measures of semantic relatedness, and thus results in better performance.

WordNet-based measures of semantic similarity suffer from shortcomings of WordNet's

topology. All similarity measures used in these experiments are unable to define a degree of semantic similarity between two words if the two words belong to different parts of speech as the concepts in different parts of speech are not linked to each other. Jarmasz and Szpakowicz (2003) mention that Hirst and St-Onge's (1998) algorithm can work with words that have different parts of speech. We tried experiments with the *hso* measure of semantic relatedness. However, we had to stop the experiment because of a very long computation time compared to the experiments with other WordNet's measures of semantic relatedness.

PMI-IR is able to calculate the value of semantic similarity between any two words, regardless of the part of speech to which they belong.

All lexicon-based measures of semantic relatedness suffer from poor coverage of words in certain domains. Many proper names and new words are not presented in the lexicon, which in turn affects the performance of lexicon-based measures of semantic relatedness. As discussed earlier, PMI-IR's performance depends on the size of document collection, which is used in the queries for calculating co-occurrence probabilities. Its performance, in the case of querying the Waterloo Multi-text System with one terabyte of unlabeled text, is significantly better than the performance of WordNet's measures of semantic relatedness.

The statistical significance test (Fisher Exact Test) on the results of experiments with WordNet's measures of semantic relatedness used in our study, demonstrated that the score of the experiments with all the measures of semantic relatedness, was not significantly better than the score of *random* measure of semantic relatedness. Some of the measures used in our experiments (such as *lch, lin*) performed well in the experiments on comparing semantic similarity measures for the Miller and Charles data (M. Jarmasz and S. Spakowicz, 2003). The Miller and Charles (1991) data consists of thirty pairs of nouns to which semantic similarity has been assigned by human experts. The explanation of such poor performance of WordNet similarity measures in our experiments could be explained by the fact that the Miller and Charles data consisted of only nouns: both words, to which the similarity measure was assigned by experts, belonged to the same

part of speech - noun. For our experiments, similarity between two words belonging to any part of speech (noun, verb and adjective) should have been assigned. But for all our WordNet-based experiments, when the two words belonged to different parts of speech, the value of zero was assigned as a similarity degree between the two words (because, as described earlier, WordNet's organization lacks links between different parts of speech). This resulted in a worse performance of WordNet similarity measures compared with their performance on the Miller and Charles data.

Another shortcoming of WordNet's similarity measures is that they perform better for nouns than other parts of speech, as WordNet's noun hierarchy is the most developed compared to other parts of speech (Patwardhan et al.,2003). Again, PMI-IR does not suffer from limitations regarding certain parts of speech.

All of the above reasons result in better performance for the experiments where semantic feature values are calculated using PMI-IR compared to the ones using WordNet-based measures of semantic relatedness.

6 Summary, Conclusions, and Future Work

This chapter summarizes the experimental study conducted for this thesis, restating the conclusions drawn from the experiments and further outlines the future work.

6.1 Summary

The goal of this thesis was an experimental study of sense ambiguity and context (specifically, the semantic context as opposed to syntactic context) for finding the optimal number of context words around a word to be disambiguated and for defining their positions from the word to be disambiguated. The study was based on modified versions of the NRC WSD system. We experimented with two different kinds of measures of semantic similarity, a corpus-based measure (PMI-IR) and several lexicon-based measures (using WordNet as the lexicon). The experiments in both cases followed the same scheme of experimenting with a different window size of content words around the word to be disambiguated.

Chapter 1 stated the problem of WSD, briefly introduced the Senseval evaluation exercise and ELS task of Senseval-2; it then presented the NRC WSD system. The goals and hypotheses of the thesis were presented next, followed by a description of the experiments of the thesis and their results.

Chapter 2 briefly introduced different concepts of context; it then reviewed statistical WSD methods such as corpus-based and knowledge-based.

Chapter 3 reviewed the ELS tasks of Senseval-2 and Senseval-3. It then described the NRC WSD system. The Machine Learning tool WEKA used in the system was presented next, followed by the discussion of two Machine Learning algorithms implemented in the system. Measures of semantic similarity (PMI-IR and WordNet's measures) were introduced last.

Chapter 4 presented the problem of context. It discussed different concepts of context used in the literature on WSD in detail and reviewed a few studies of ambiguity and context. The hypotheses of the thesis were restated next, followed by the methodology for testing each hypothesis.

Chapter 5 presented results of the experiments for testing the hypotheses of the thesis together with the results of statistical significance tests. The chapter concluded with the discussion of results and a comparison of the results with related studies.

Chapter 6 presents a summary of the thesis, presents conclusions and observations of our study as well as related studies of ambiguity and context, discussed in the thesis. It concludes with presenting some ideas for future work.

6.2 Conclusions

This thesis has shown that the window size of content words immediately preceding and following the ambiguous word must be relatively small to be useful in WSD systems of the kinds that we investigated.

The initial hypothesis of this thesis, that the accuracy of the modified NRC WSD system will improve as more semantic features are introduced, by expanding the window of content words around the head word, for our experiments is not supported. The score of the experiments with a wider window of content words (four semantic features) around the head word was not significantly different from the score of the experiments with a small window of content words (two nearest semantic features). However, taking into account the computation speed, the experiments with the two semantic features had the best performance.

Thus, the optimal number of content words for our modified version of the NRC WSD system for the ELS task of Senseval-2, is two words. The distance of the context words with respect to the target word plays a significant role. For our experiments, a window of

one content word immediately preceding, and one content word immediately following the target word, resulted in the best performance of the system.

The second hypothesis aimed at investigating why the system's performance did not improve when more content words were taken into account. We hypothesized that the additional content words (and corresponding semantic features) were less relevant to disambiguating the target word. Our experiments demonstrated that the two distant words were less relevant compared with the two nearest words. The system's performance decreased significantly in the case of two furthest words compared to the two nearest words.

An alternative hypothesis was that the additional features were relevant but redundant. The result of the experiments with the two furthest features showed that the features were not redundant. If they were redundant, the score of the experiments with the two nearest and two furthest semantic features would have been the same. Thus, the hypothesis about additional features being less relevant for disambiguation, is supported.

The purpose of the third hypothesis was to investigate how general our results, obtained in testing the previous two hypotheses, were. We were interested in experimenting with different measures of semantic relatedness in place of PMI-IR and observing the performance of the system. We hypothesized that the results should be similar to the experiments with PMI-IR measure of semantic similarity: the first hypothesis should be rejected and the second hypothesis should be accepted.

The experiments for testing the third hypothesis, demonstrated that the results of WordNet-based measures of semantic relatedness follow the same pattern with the results of the experiments with PMI-IR: the Hypothesis 1 was rejected and the Hypothesis 2 was accepted. These observations suggest that the results of the experiments are not specific to a statistical measure of semantic similarity (PMI-IR) or lexicon-based measures of semantic relatedness (WordNet measures).

Another observation from the experiments using WordNet's measures of semantic relatedness, is their poor performance. Performance of all the measures used in our

experiments was not significantly better than performance of *random* measure. As noted earlier, WordNet's shortcomings (e.g., poor coverage of words and lack of links between semantically related concepts in different parts of speech) resulted in the measures' poor performance compared with PMI-IR and with the experiments on the Miller and Charles data (Jarmasz and Szpakowicz, 2003).

As discussed in the thesis, the problem of ambiguity and context depends on many factors. Disambiguation of different ambiguous words (such as homonyms and polysemous words) and words belonging to different parts of speech require different window sizes (e.g., short context and long context) and different types of context (e.g., syntactic and semantic). Some words benefit from domain information in wide contexts; for others, domain information is less useful, and they can be disambiguated only by local context. The disambiguation results depend on the type of features used in context (e.g., collocations and syntactic relations). Thus, there is no universal solution for the problem of WSD and context. However, some common trends in research on the topic can be used for further investigation of the problem.

To conclude, even though larger contexts intuitively seem to be more useful for WSD, in practice, for the majority of cases, the words in close proximity to the target word are the most beneficial in micro-context. The sentence content seems to be the most beneficial for deciding on the topic of the ambiguous word. Domain information is more accurate when portions of the text are considered instead of the whole document. The hypothesis drawn from these observations is that wider contexts bring in more information for WSD, but they also bring more noise.

6.2.1 Conclusions from Studies on WSD and Context

The amount of context taken into account plays a significant role for WSD. As seen from the sections reviewing systems using different context for WSD, it is one of the central problems for word sense ambiguity resolution. We present some of the observations and

conclusions drawn from the discussed studies. WSD success and context depend on many factors, which we believe, should be considered and investigated in further research on WSD and context. Some of the factors are:

- Substantive words are found to be useful for topical and domain context whereas stop words are useful for local context;
- 1- or 2-contexts from the close neighborhood of a target word are sufficient for WSD in micro-context in majority of cases. Wider contexts within a sentence add to noise;
- Different parts of speech might require different contexts. Nouns seem to benefit from topical context, whereas verbs and adjectives benefit from local context (Yarowsky, 1993);
- Different types of ambiguity might require different contexts: homonyms seem to be disambiguated from larger contexts and polysemous words from shorter context (Crestan, 2004);
- WSD results depend on the training corpora. (Semcor is better (Agirre et al., 2001) for topic signatures than the DSO corpus);
- There is a correlation between the size of context required for WSD and the number of samples and distribution of senses in training data (Lamjiri, 2004);
- For domain-driven WSD, taking into account a portion of the text instead of the whole text of the document is more useful for WSD. WSD by using domain information requires a large context (about 100 content words) and decreases when fewer words are taken into account (Magnini et al., 2002);
- Words that do not belong to any domain (factotum words) do not benefit from domain information and should be disambiguated by using local context (Magnini et al.,2002);
- There is a correlation between WSD results and the Machine Learning algorithms implemented in the systems;
- WSD results depend on the lexicon used: most of the systems use WordNet senses (though some of them try to enrich WordNet with topic signatures and domain labels); the training corpora are also labeled with WordNet senses.

Some of the shortcomings of WordNet are poor coverage of words in some domains, which might result in poor WSD results.

In summary, the problem of ambiguity and context remains an open problem. More studies aimed at investigating the role of context in WSD need to be carried out.

Different interpretations of local and topical context by the authors of systems performing WSD, and also different features used for Machine Learning, present some difficulties for comparing such systems and for finding regularities in sense ambiguity resolution and considered context.

6.3 Future Work

This section attempts to outline some directions for future work with the NRC WSD system:

1) As mentioned in the Section 5.4.1, one of the possibilities to explain that a wide context was beneficial in studies on topic signatures and domains, is that context features in those systems were based on properties of a group of words and not on individual words. This hypothesis will be tested in future work on the NRC WSD system.

2) The system's performance could be improved by having a feature selection for the classifier, which would take into account semantic similarity of a word from context with the head word. For instance, similar to the Brown et al. (1991) approach, we could select the best feature and disambiguate based on that single feature. The system could use PMI-IR to select a single feature; the most semantically relevant content word from the four words around it in the context. In the 'information theory' approach (Brown et al.,1991), mutual information is defined using an annotated corpus. PMI-IR uses a collection of unlabeled texts for that purpose. In our experiments, we also experimented with one feature (it was a content word located in the position immediately preceding the head word). However, we did not select the most semantically relevant content word

among the four words around the head word.

3) The NRC WSD system used an evaluation of semantic features: semantic features with values less than a certain threshold value were dropped. Similarly, we could select features using the 'information theory' approach. For example, while choosing the semantic features we could have some prescreening of the features: selecting the ones with higher mutual information with the head word and skipping the ones which are less relevant, until we have extracted the required number of words from data (for example, four words, as it was in the experiments with wide context in our experiments). This could improve the system's performance, as the irrelevant features (or noise) would not be considered while training, and thus the chances for over-fitting would be decreased.

4) Another thing to test could be the use of context as a bag of words. As noted earlier, systems using topical contexts use the bag of words approach. For the experiments described in this thesis, the relational approach was used, where the position of a context word was fixed. For the bag of words approach, a head word would have a certain number of semantic features (context words) without limiting them to a certain position. Content words semantically close to the head word may be located in different positions in the sentences of the training and test data. When calculating semantic feature values, we expect this approach would require more computation and thus would be more time consuming than the relational approach used in the system. Semantic feature values would be evaluated by finding the similarity between each word from four words (if the number of words is four as it was in one of our experiments) taken into account in the training example with all four words in the test example.

5) For this thesis, we experimented only with one type of context for semantic features: topical context (though it wasn't purely topical). Possibilities for future work could involve experimenting with only micro-context and domain context (as defined by Ide and Veronis, 1998).

6) It would also be interesting to repeat the experiments with a different number of words around the head word while slightly modifying the system so that the semantic features

would be limited to the sentence boundaries as in the original version of the NRC WSD. The idea here is that more content words would be extracted from the sentence of the target word and if some content words were missing, special null characters would be inserted in place of a missing word.

7) Crestan (2004) hypothesized that homonyms seem to benefit from wider contexts (more context words around the head word) compared to short contexts, while disambiguation of polysemous words tends to worsen in the case of larger contexts. As most of the words tested for these experiments were polysemous, this hypothesis could be verified on the data containing homonyms.

8) The NRC WSD system was tested on the ELS task. Testing its performance on ELS tasks of different languages would give a possibility to explore whether ambiguity and context have similar trends in different languages. Another application of the system could involve an 'all words' task.

9) Recent systems (e.g. Agirre and Martinez, 2004) combine local, topical, and domain features to improve WSD. In future work, we could experiment with different contexts and their combinations to find which contexts result in the best performance of the system.

10) Many systems accomplishing WSD use a variety of features of the classifiers. A possibility to improve our modified NRC WSD system's performance could be adding grammatical features, which would reflect the grammatical structure of the sentence such as object-verb, verb-subject relationships.

11) Finally, the NRC WSD system could use a set of expanded features similarly to (Cabezas et al., 2004). In their system, 'expanded context' was obtained by Information Retrieval. They used words in local context as a bag of words to query an Information Retrieval system to obtain non-stop words from *n* top documents retrieved. Those words were included into features of the classifier. We could query WMTS to obtain the similar documents and use the additional words as extended features for the classifier.

In summary, we repeat, even within the context of statistical approaches to WSD, the problem of ambiguity and context remains open.

References

1. Agirre E., Martinez D. (2004). The Basque Country University system: English and Basque tasks, In *Proceedings of SENSEVAL-3 Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 25-26 July, Barcelona, Spain, 44-48.

2. Agirre, E., Ansa, O., Martinez, D., Hovy, E. (2001). Enriching WordNet concepts with topic signatures. In *Proceedings of the NAACL workshop on WordNet and Other Lexical Resources*: Applications, Extensions and Customizations. Pittsburg, USA.

3. Agirre E., Martinez D. (2000). Exploring automatic word sense disambiguation with decision lists and the Web, http://arxiv.org/PS_cache/cs/pdf/0010/0010024.pdf.

4. Agresti, A. (1990). Categorical Data Analysis, Wiley and Sons Publishers.

5. Brown P., Pietra, S., Pietra, V., Mercer, R. (1991). Word Sense Disambiguation using Statistical Methods. In *ACL* (29), 264-270.

6. Choueka Y., Lusignan S. (1985). Disambiguation by short context. *Computers and the Humanities,* 19, 147-158.

7. Corriveau, J.-P. (1995). Time Constrained Memory. Lawrence Erlbaum Associates Publishers, Mahwah, New Jersey, Hove, UK.

8. Crestan, E. (2004). Contextual Semantics for WSD, In *Proceedings of SENSEVAL-3 Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 25-26 July, Barcelona, Spain, 101-104.

9. Dagan I., Itai, A.(1994). WordSense Disambiguation Using a Second Language Monolingual Corpus, *Computational Linguistics 20(4)*, 563-596.

10. Decadt B., Hoste V., Daelemans W. (2004). GAMBL, Genetic Algorithm Optimization of Memory-Based WSD, In *Proceedings of SENSEVAL-3 Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 25-26 July, Barcelona, Spain, 108-112.

11. Fellbaum, C. (1997). WordNet: An Electronic Lexical Database and Some of its Applications. *MIT Press, Cambridge.*

12. Gonzalo J., Chugur I., Verdejo F. (2003). The web as a resource for WSD, http://ixa.si.ehu.es/Ixa/local/meaning-workshop/papers/julio.pdf

13. Hirst, G. and St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In Fellbaum, C., ed., *WordNet: An electronic lexical database.* MIT Press. 305-332.

14. Ide, N. and Veronis, J. (1998). Word Sense Disambiguation: The state of the Art. *Computational Linguistics*, 24(1).

15. Jarmasz, M. and Szpakowicz, S. (2003). Roget's Thesaurus and Semantic Similarity. In *Proceedings of Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, Borovets, Bulgaria, September, 212-219.

16. Jiang, J., Conrath, D. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy, In *Proceedings of International Conference Research on Computational Linguistics.*

17. Kaplan A. (1950). An Experimental study of ambiguity and context. Published as: Kaplan, Abraham (1955). An experimental study of ambiguity and context. *Mechanical Translation,* **2**(2), 39-46.

18. Kawamoto, A. H. (1993). Nonlinear Dynamics in the Resolution of Lexical ambiguity: A Parallel Distributed Processing Account. *Journal of Memory and Language, 32,* 474-516.

19. Kilgarriff, A., Palmer, M. (2000). Special issue on SENSEVAL. *Computers and Humanities,* 34 (1-2).

20. Kilgarriff (1998). SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs. *In Procedeengs LREC*, Granada, 581—588.

21. Kohomban U. (2005). Addressing the Knowledge Acquisition Bottleneck in Large Scale WSD,
http://wing.comp.nus.edu.sg/chime/textSeminar.html

22. Koutsoudas A., Korfhage R. (1956). M.T. and the problem of Multiple Meaning. *Mechanical Translation,* 2(2), 46-51.

23. Lamjiri A., Demerdash O., Kosseim L. (2004). Simple Features for Statistical WSD, In *Proceedings of SENSEVAL-3 Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 25-26 July, Barcelona, Spain, 133-136.

24. Leacock, C., Miller, G., Chodorow, M. (1998). Using Corpus statistics and WordNet Relations for Sense Identification. *Computational Linguistics,* 24(1).

25. Leacock, C., and Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. In Fellbaum, C., ed., *WordNet: An electronic lexical database.* MIT Press, 265-283.

26. Lee Y., Ng H., Chia, T. (2004). Supervised WSD with Suport Vector Machines and Multiple Knowledge Sources, In *Proceedings of SENSEVAL-3 Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 25-26 July, Barcelona, Spain, 137-140.

27. Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the International Conference on Machine Learning.*

28. Lewis, D (1991). Evaluating Text Categorization, In *Proceedings of Speech and Natural Language Workshop.*

29. B. Magnini, C. Strapparava, G. Pezzulo, A. Gliozzo (2002). The Role of Domain Information in Word Sense Disambiguation, *Natural Language Engineering*, special issue on Word Sense Disambiguation, 8(4), pp. 359-373, Cambridge University Press.

30. Manning, C.D. and Schuetze, H. (1999). Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge, Massachusetts, London, England.

31. Mihalcea, R. and Moldovan, (1999). An Automatic Method for Generating Sense Tagged Corpora, in *Proceedings of the American Association for Artificial Intelligence,* Orlando, FL.

32. Mihalcea, R. (2004). Co-training and Self-training for Word Sense Disambiguation. In Proceedings of the Conference on Natural Language Learning.

33. Mihalcea, R., Chklovski, T., Kilgarriff A. (2004). The SENSEVAL-3 English Lexical Sample Task. In *Proceedings of SENSEVAL-3 Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 25-26 July, Barcelona, Spain, 25-28.

34. Miller, G. (1995). WordNet: A Lexical Database. *Communication of the ACM,* 38 (11):39-41.

35. Miller, G. and Charles, W. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1-28.

36. Norvig, P. (1989). Marker Passing as a Weak Method for Text Inferencing. *Cognitive Science* 13(4): 569-620.

37. Ng, H., Lee H. (1996). Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-based Approach. *In ACL Proceedings.*

38. Paliouras G., Karkaletsis V., Androutsopoulos I., Spyropoulos C.(2000). Learning Rules for Large-Vocabulary Word Sense Disambiguation: A Comparison of Various Classifiers. *Natural Language Processing*, 383-394.

39. Pathwardan, S., Satanjeev, B and Pedersen, T. (2003). Using Measures of Semantic Relatedness for Word Sense Disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics,* Mexico City, 241-257.

40. Pedersen T., Pathwardhan S., and Michelizzi J. (2004). WordNet::similarity – measuring the relatedness of concepts. In *Proceddeings of Fifth Annual Meeting of the North American Chapter of the ACL (NAACL-04,* Boston, MA.

41. Pustejovsky, J. (1991). The Generative Lexicon. Computational Linguistics 17(4): 409-441.

42. Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 448–453.

43. Ram, A. and Moorman, K. (1999). Understanding Language Understanding. MIT Press, Cambridge, Massachusetts.

44. Small, S.L. (1980). Word Expert Parsing: A Theory of Distributed Word-based Natural Language Understanding, The University of Maryland, Baltimore.

45. Terra, E. and Clarke, C. (2003). Frequency Estimates for Statistical Word Similiarity Measures. *Proceedings of the Human Language Technology Conference*, Edmonton, Canada.

46. Turney, P. (2004). Word Sense disambiguation by Web Mining for Word Co-occurrence Probabilities. In *Proceedings of SENSEVAL-3 Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 25-26 July, Barcelona, Spain, 239-242.

47. Turney, P. (2003). Coherent Keyphrase Extraction via Web Mining, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03).* Acapulco, Mexico. August 9-15, 2003. pp. 434-439.

48. Turney, P. (2001). Mining the Web for Synonyms: PMI-IR vs LSA on TOEFL. *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, Freiburg, Germany, pp. 491 – 502.

49. Weaver, W. (1949). *Translation.* Mimeographed, July 15, 1949. Reprinted in Locke, William N. and Booth, A. Donald (1955) (Eds.), *Machine translation of languages.* John Wiley & Sons, New York, 15-23.

50. Weiss, S. and Kulikowski, C. (1991). Computer Systems That Learn. Morgan Kaufmann Publishers, San Mateo, California.

51. Witten, I. and Frank, E. (1999). Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, San Francisco, California.

52. Wu, Z., and Palmer, M. (1994). Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, 133–138.

53. Yarowsky, D. (1994a), Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. In *Proceedings of the 32nd conference on Association for Computational Linguistics*, 88-95.

54. Yarowsky, D. (1994b), A Comparison of Corpus-based Techniques for Restoring Accents in Spanish and French Text. *Proceedings of the Second Annual Workshopon Very Large Text Corpora,* 19-32.

55. Yarowsky, D. (1993), One Sense per Discourse, One Sense per Collocation. In *Proceedings of ARPA Human Language Technology Workshop,* Princeton, New Jersey, 266-271.

56. Yarowsky, D. (1992), Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. *COLING,* 454-460.

Appendix A: Score of the experiments with PMI-IR

		Pointwise Mutual Information - Information Retrieval											
Word	Test	4 features		3 features		2 nearest		2 furthest		1 feature		0 features	
	size	f	c	f	c	f	c	f	c	f	c	f	c
art	98	69.4	72.4	71.4	75.5	79.6	81.6	59.2	60.2	74.5	76.5	55.1	57.1
authority	92	73.9	88.0	77.2	89.1	76.1	89.1	77.2	84.8	72.8	89.1	77.2	89.1
bar	151	60.3	67.5	64.2	70.9	68.2	74.8	63.6	70.2	67.5	76.2	47.7	53.6
bum	45	77.8	80.0	77.8	80.0	77.8	80.0	77.8	80.0	77.8	77.8	77.8	80.0
chair	69	85.5	85.5	88.4	88.4	88.4	88.4	85.5	85.5	87.0	87.0	84.1	84.1
channel	73	57.5	63.0	50.7	58.9	58.9	65.8	43.8	52.1	52.1	60.3	41.1	52.1
child	64	76.6	76.6	70.3	70.3	73.4	73.4	81.3	81.3	75.0	75.0	73.4	73.4
circuit	85	71.8	74.1	72.9	75.3	68.2	69.4	69.4	71.8	65.9	68.2	56.5	56.5
fatigue	43	83.7	88.4	86.0	90.7	86.0	90.7	79.1	83.7	86.0	90.7	81.4	86.0
feeling	51	64.7	64.7	58.8	58.8	64.7	64.7	66.7	66.7	66.7	66.7	70.6	72.5
nouns	771	70.3	74.9	70.7	75.4	73.3	77.5	68.8	72.5	71.3	76.5	63.0	67.4
use	76	68.4	81.6	69.7	82.9	68.4	81.6	69.7	82.9	67.1	81.6	68.4	82.9
wander	50	82.0	90.0	82.0	90.0	82.0	90.0	82.0	90.0	82.0	90.0	78.0	90.0
wash	12	33.3	50.0	41.7	50.0	33.3	50.0	33.3	50.0	33.3	50.0	33.3	50.0
work	60	41.7	48.3	40.0	50.0	45.0	50.0	31.7	41.7	38.3	51.7	45.0	56.7
call	66	37.9	56.1	37.9	57.6	37.9	57.6	40.9	63.6	42.4	59.1	47.0	65.2
carry	66	43.9	50.0	45.5	51.5	42.4	50.0	36.4	43.9	37.9	45.5	30.3	39.4
collaborate	30	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0	80.0	80.0
develop	69	37.7	59.4	37.7	59.4	37.7	56.5	34.8	56.5	30.4	47.8	33.5	52.2
draw	41	22.0	31.7	26.8	41.5	24.4	34.1	22.0	36.6	29.3	34.1	26.8	34.1
keep	67	55.2	56.7	56.7	58.2	55.2	56.7	58.2	59.7	56.7	58.2	55.2	56.7
verbs	537	51.2	61.6	52.1	63.3	51.6	61.8	49.7	61.6	50.3	60.7	49.9	61.3
blind	55	90.9	90.9	85.5	85.5	85.5	85.5	89.1	89.1	85.5	85.5	89.1	89.1
cool	52	65.4	65.4	67.3	67.3	65.4	65.4	59.6	59.6	57.7	57.7	48.1	48.1
faithful	23	82.6	82.6	82.6	82.6	78.3	78.3	78.3	78.3	78.3	78.3	82.6	82.6
fine	70	61.4	61.4	62.9	62.9	61.4	61.4	48.6	48.6	54.3	54.3	42.9	42.9
free	82	65.9	65.9	63.4	63.4	65.9	65.9	46.3	46.3	51.2	51.2	42.7	42.7

graceful	29	79.3	79.3	79.3	79.3	82.9	82.9	79.3	79.3	79.3	79.3	79.3	79.3
green	94	87.2	87.2	87.2	87.2	89.4	89.4	84.0	84.0	87.2	87.2	83.0	83.0
local	38	68.4	68.4	68.4	68.4	71.1	71.1	60.5	60.5	60.5	60.5	57.9	57.9
natural	103	62.1	62.1	65.0	65.0	63.1	63.1	49.5	49.5	45.6	45.6	48.5	48.5
simple	66	57.6	57.6	57.6	57.6	51.5	51.5	59.1	59.1	51.5	51.5	56.1	56.1
adjectives	612	70.7	70.7	70.7	70.7	70.3	70.3	62.9	62.9	62.7	62.7	60.1	60.1
all	1920	65.1	69.9	65.5	70.5	66.3	70.8	61.6	66.4	62.7	67.7	58.5	63.4

Appendix B: Statistical significance tests on PMI-IR experiments' results

test sample	sample size	4 features		2 nearest		difference (2-4)		p-value		95% significance	
		f	c	f	c	f	c	f	c	f	c
noun	771	70.3	74.9	73.3	77.5	3.0	2.6	0.106564	0.11584	NO	NO
verb	537	51.2	61.6	51.6	61.8	0.4	0.2	0.47566	0.499999	NO	NO
adjective	612	70.7	70.7	70.3	70.3	-0.5	-0.5	0.450125	0.450125	NO	NO
all	1920	65.1	69.9	66.3	70.8	1.1	0.9	0.227266	0.285947	NO	NO

test sample	sample size	3 features		2 nearest		difference (2-3)		p-value		95% significance	
		f	c	f	c	f	c	f	c	f	c
noun	771	70.7	75.4	73.3	77.5	2.6	2.1	0.140638	0.168429	NO	NO
verb	537	52.1	63.3	51.6	61.8	-0.5	-1.5	0.451397	0.329485	NO	NO
adjective	612	70.7	70.7	70.3	70.3	-0.4	-0.4	0.450125	0.450125	NO	NO
all	1920	65.5	70.5	66.3	70.8	0.8	0.3	0.316818	0.443638	NO	NO

test sample	sample size	2 furthest		2 nearest		difference (2-2)		p-value		95% significance	
		f	c	f	c	f	c	f	c	f	c
noun	771	68.8	72.5	73.3	77.5	4.5	5.0	0.028139	0.012646	YES	YES
verb	537	49.7	61.6	51.6	61.8	1.9	0.2	0.291411	0.499999	NO	NO
adjective	612	62.9	62.9	70.3	70.3	7.4	7.4	0.003816	0.003816	YES	YES
all	1920	61.6	66.4	66.3	70.8	4.7	4.4	0.001385	0.001752	YES	YES

test sample	sample size	1 feature		2 nearest		difference (2-1)		p-value		95% significance	
		f	c	f	c	f	c	f	c	f	c

noun	771	71.3	76.5	73.3	77.5	2.0	1.0	0.212809	0.335849	NO	NO
verb	537	50.3	60.7	51.6	61.8	1.3	1.1	0.357105	0.377077	NO	NO
adjective	612	62.7	62.7	70.3	70.3	7.6	7.6	0.003195	0.003195	YES	YES
all	1920	62.7	67.7	66.3	70.8	3.6	3.1	0.010906	0.021258	YES	YES

test sample	sample size	0 features		2 nearest		difference (2-0)		p-value		95% significance	
		f	c	f	c	f	c	f	c	f	c
noun	771	63.0	67.4	73.3	77.5	10.3	10.1	0.000009	0.000005	YES	YES
verb	537	49.9	61.3	51.6	61.8	1.7	0.5	0.312687	0.450089	NO	NO
adjective	612	60.1	60.1	70.3	70.3	10.2	10.2	0.000123	0.000123	YES	YES
all	1920	58.5	63.4	66.3	70.8	7.8	7.4	0	0	YES	YES